PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q 11/04, H04L 12/56	A1	(11) International Publication Number: WO 98/30059 (43) International Publication Date: 9 July 1998 (09.07.98)

(21) International Application Number: PCT/CA97/00991

(22) International Filing Date: 31 December 1997 (31.12.97)

(30) Priority Data:
60/034,607 3 January 1997 (03.01.97) US

(71) Applicant *(for all designated States except US)*: TELECOMMUNICATIONS RESEARCH LABORATORIES [CA/CA]; 800 Park Plaza, 10611-98 Avenue, Edmonton, Alberta T3K 2P7 (CA).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: RUEDA, Jose, A. [CA/CA]; Unit 10, 31 Carlton Street, Winnipeg, Manitoba R3C 1N71 (CA). BOURLAS, Yair [CA/CA]; 35 Leamington Gate, Winnipeg, Manitoba R3Y 1B8 (CA). DIAMOND, Jeffrey, E. [CA/CA]; 8-812 Wolseley Avenue, Winnipeg, Manitoba R3G 1C6 (CA). McLEOD, Robert, D. [CA/CA]; 38 Elsbury Avenue, Winnipeg, Manitoba R2N 2C4 (CA).

(74) Agent: ADE & COMPANY; Patent and Trademark Agents, 1700 – 360 Main Street, Winnipeg, Manitoba R3C 3Z3 (CA).

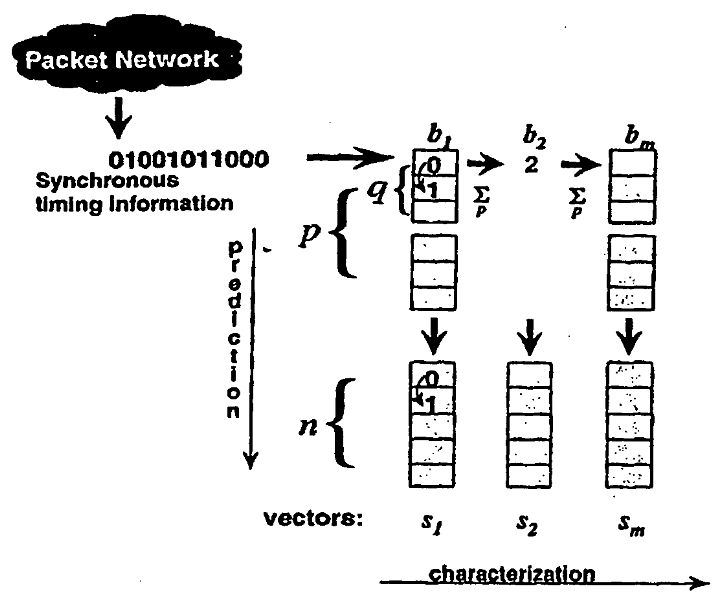
(81) Designated States: CA, US, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD FOR REAL-TIME TRAFFIC ANALYSIS ON PACKET NETWORKS

(57) Abstract

An architecture for capture and generation, and a set of methods for characterization, prediction, and classification of traffic in packet networks are disclosed. The architecture consists of a device that stores packet timing information and processes the data so that characterization, prediction, and classification algorithms can perform operations in real-time. A methodology is disclosed for real-time traffic analysis, characterization, prediction, and classification in packet networks. The methodology is based on the simultaneous aggregation of packet arrival times at different times cales. The traffic is represented at the synchronous carrier level by the arrival or non-arrival of a packet. The invention does not require knowledge about the information source, nor needs to decode the information contents of the packets. Only the arrival timing information is required. The invention provides a characterization of the traffic on packet networks suitable for a real-time implementation. The methodology can be applied in real-time traffic classification by training a neural network from calculated second order statistics of the traffic of several known sources.



Performance descriptors for the network can also be obtained by calculating the deviation of the traffic distribution from calculated models. Traffic prediction can also be done by training a neural network from a vector of the results of a given processing against a vector of results of the subsequent processing unit; noticing that the latter vector contains information at a larger time scale than the previous. The invention also provides a method of estimating an effective bandwith measure in real time which can be used for connection admission control and dynamic routing in packet networks. The invention provides appropriate traffic descriptors that can be applied in more efficient traffic control on packet networks.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Method for Real-Time Traffic Analysis on Packet Networks

The Field of the Invention:

This invention relates in general to traffic analysis on packet networks. Packet networks are telecommunication networks in which the information is transmitted in small binary groups called packets. An advantage of packet networks is that it can handle different sources simultaneously by processing the packets sequentially. The packets travel through the network via a fast synchronous carrier; this can be viewed as a train of pulses which transport the packets. The speed of this carrier determines the maximum speed of the packet network. These networks can process only one packet at a time and thus the order in which the packets are processed depends on the priorities and quality of service required by the sources. Packets from a given source are mixed with packets from other sources. Each packet has a header that identifies its destination. Once the packets arrive at their destination, the headers are removed and the information is reassembled. Since these networks generally carry traffic from different types of sources which demand different service levels from the network, it is important from a network operation and management point of view to know the characteristics of the expected traffic. In the standards for packet networks that use the asynchronous transfer mode protocol (ATM), the traffic is described by first order statistics such as the peak cell rate, cell delay variation tolerance, sustainable cell rate, and maximum burst size. The size of the packets in ATM networks (that is networks which utilize the ATM protocol) is 53 bytes, of which 5 bytes compose the header and the remaining 48 bytes contain a section of the information being transmitted. The rate is defined as the number of packets that flow through the network in a given unit of time; this is a measure of the speed of the telecommunications network. The peak cell rate is defined as the inverse of the minimum time between successive packet arrivals to a switch. The cell delay variation tolerance is defined as the sensitivity of the information to changes in delay of the packets as they flow through the network. The sustainable cell rate is defined as the maximum average rate. The maximum burst size is defined as the maximum number of cells at the peak rate. Two of these parameters, the peak cell rate and the sustainable cell rate, have been defined as mandatory traffic parameters (or descriptors) in the ATM Forum UNI version 3.0 standards, as explained by McDysan and Spohn [11].

There are two types of networks which differ according to the way a connection is handled. The first type is connection-oriented. In these networks it is required to set up several parameters before any data transmission can take place. This is a process in which a source negotiates a level of service with the network. An end-to-end path with a quality of service is established and all the packets from the source will follow this path.

The second type is connectionless in which it is not required to set up an end-to-end connection. The network handles the packets individually.

Quality of service (QoS) is a parameter meaningful from a source to a destination point of view, as well as at each link in the network. In connection-oriented networks, the negotiation for a QoS is carried out by agreeing on certain parameters. These parameters are based on first order statistical measures of the performance, such as the average delay, cell delay variation, error rates, and different levels of packet loss, as explained in the Bellcore requirements for broadband switching systems [2].

Packet networks have been designed to carry traffic from multimedia sources, among them different types of video and audio, voice, and data communications. Each traffic source presents the network with different requirements. The network must be able to handle all these traffic sources at their respective quality of service. The problem that arises is how to accurately characterize the different traffic sources for efficient network utilization. It might, in fact, be required to measure quality in different ways for different traffic sources. This makes the performance measurement problem very complex.

In our invention, a method is used to characterize the traffic in real-time. The method is used to calculate traffic descriptors considering properties of the traffic which have not been considered previously in commercial equipment. The descriptors are based on properties of the traffic that have been reported in the literature; but the techniques available are not suitable for real-time measurements. The algorithm presented is based on the simultaneous measurement of the traffic at different time scales. The data is represented in an appropriate form, processed and organized in an array of vectors. From this array, higher order statistical measures are derived. The traffic descriptors calculated in this way are used to characterize the traffic. The algorithm is implemented in real-time. The information is also used for traffic classification and performance prediction.

The Description of Related Art:

The following references have been identified in a search in this field, some of which are relevant to the present invention:

Publications

[1] R. Addie, M. Zukerman, and T. Neame, "Fractal Traffic: Measurements, Modelling and Performance Evaluation", in Proc. IEEE Infocom, pp. 977-984, 1995.

[2] Bellcore, "Broadband Switching System Generic Requirements", GR-1110-CORE, Revision 3, April 1996.

[3] J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-Range Dependence In Variable-Bit-Rate Video Traffic", in IEEE Trans. on Communications, vol. 43, no. 4, pp. 1566-1579, April 1995.

[4] Y. Chen, Z. Deng, and C. Williamson, "A Model for Self-Similar Ethernet LAN Traffic: Design, Implementation, and Performance Implications", internal report, University of Saskatchewan, Canada, 1995.

[5] M. Devetsikiotis, I. Lambadaris, R. Kaye, "Traffic Modeling and Design Methodologies for Broadband Networks", *Canadian Journal on Electrical and Computer Engineering*", vol. 20, no. 3, 1995.

[6] M. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic", in Proc. ACM Sigcom, London, UK, pp. 269-280, 1994.

[7] R. Guerin, H. Ahmadi and M. Naghshineh, "Equivalent Capacity and its to Bandwidth Allocation in High Speed Networks", *IEEE JSAC,* vol. 9, no. 7, 1991.

[8] C. Huang, M. Devetsikiotis, I. Lambadaris, and A. Kaye, "Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach", in ACM Sigcom, Cambridge 1995.

[9] W. Lau, A. Erramilli, J. Wang, and W. Willinger, "Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and its Properties", in Proc. IEEE Int Conf. Commun., 1995.

[10] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On The Self-Similar Nature of Ethernet Traffic (extended version)", *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 1-15, February 1994.

[1] B. Mandelbrot, "Self-Similar Error Clusters in Communication Systems and the Concept of Conditional Stationarity", in IEEE Trans. on Communication Technology, pp. 71-90, 1965.

[12] D. McDysan and D. Spohn, "ATM Theory and Application". Toronto: McGraw-Hill, 1995.

[13] D. McLaren and D. Nguyen, "A Fractal-Based Source Model for ATM Packet Video", in Int. Conf. on Digital Processing of Signals in Communications, Univ. of Loughbovough, September 1991.

[14] V. Paxson, "Fast Approximation of Self-Similar Network Traffic", report LBL-36750, Univ. of California at Berkeley, Lawrence Berkeley Laboratory, 1995.

[15] A. Rueda and W. Kinsner, "A Survey of Traffic Characterization Techniques in Telecommunication Networks", *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 830-833, May 1996.

<u>U.S. Patents</u>

5,050161 Congestion management based on multiple framing strategy

5,274,625 Traffic measurements in packet communication networks

5,341,366 Connection admission control system

5,343,463 Performance measurement system for a telecommunication path and device used therein

5,343,465 Method and system for real-time burstiness analysis of network traffic

5,357,507 Fast connection admission control for ATM networks

5,357,510 Apparatus and method for supervising and controlling ATM traffic

5,365,514 Event driven interface for a system for monitoring and controlling a data communications network

5,375,070 Information collection architecture and method for a data communications network

5,394,394 Message header classifier

5,426,635 Method for adaptive control of windows and rates in networks

5,434,848 Traffic management in packet communication networks

5,448,567 Control architecture for ATM networks

The traffic characterization techniques for telecommunication networks found in the literature can be classified into the following categories [5], [14]: autoregressive moving average (ARMA) models, Bernoulli process modeling, Markov chain modeling, neural network models, self-similar models, transform-expand-sample (TES) models, traffic flow models, and wavelet models.

Different kinds of stochastic (statistical) models reported in the literature have successfully been used in modeling traffic in telecommunication networks. For example, Markov chains are a useful tool in modeling communication systems. It is widely accepted that the short-term arrival processes in telecommunication networks can be accurately described by Poisson processes, for example an FTP control connection which can be modeled as a Markov modulated Poisson process (MMPP) [13].

Traffic on packet networks is irregular in nature as explained by Leland, et al. [9]. It is generally accepted that the long-range dependencies found in multimedia traffic can be described using models which consider self-similarity. Self-similarity is a measure of the variation of the traffic properties at different time scales. Several traffic models of this type have been reported [1], [4], [6], [7], [8], and [12].

The traffic descriptors that have been utilized in packet networks are based on statistical measures such as the mean, peak and sustained rates, burst length, and cell-loss ratios. These do not quantify correlation well, and thus a need exists for descriptors that provide more information in order to describe highly correlated and bursty (irregular) multimedia traffic [3] and [10].

Traditional analysis techniques cannot be successfully applied. The methods that have been proposed for the analysis of traffic in packet networks are impractical for a real-time implementation since they required calculations that involved several processing phases on a stored time series.

A method for time deviation (TDEV) calculation for synchronization monitoring in SONET/SDH networks is proposed by Grover and Stamatlakis [15]. Their method consists of the calculation of simultaneous averages of the second differences of time interval errors (TIE) at different time scales. A time scale is represented by blocks that contain a number of second differences which is a power of two. For each block the second differences are added and the result is squared. The sums of the squares are then divided by six and by a constant to obtain a measure called the time variance (TVAR) for the time scale. Each new TIE value produces a new second differences and this is used to updated all the block sums of all the time scales. Each new second difference could complete a block sum for one or more blocks and a new TVAR could be obtained. The results are used to produce a plot of the logarithm base two of the TVAR values of all the blocks

versus the logarithm base two of the block size. Their paper also details the standard block calculation which consists of calculating the TVAR values for all the blocks in a batch mode (or block mode, or off-line). This consists of calculating the block sums and TVAR values of a time series of TIE values for each block separately.

[7] W. D. Grover and D. Stamatlakis, "Continuous TDEV calculation for in-situ synchronisation monitoring in SONET/SDH networks", *Electronic Letters*, vol. 29, No. 16, pp. 1405-1406, August 1993.

Namajunas and Tamasevicius [16] proposed a device for measuring fractal dimensions from a time series in real-time. Their device is an electronic circuit that produces an approximation to the fractal dimension of a class of analog electric signals in real-time.

[8] A. Namajunas and A. Tamasevicius, "A technique for measuring fractal dimensions from time series on a real-time scale", *Physica D*, vol. 58, pp. 482-488, 1992.

SUMMARY OF THE INVENTION

It is one object of the present invention to provide a method for characterizing transmissions in a packet network which may or may not be carried out in real time and may provide information which can be used to characterize the transmissions for prediction and access control.

According to a first aspect of the invention there is provided a method for deriving information related to characteristics of transmissions in a packet network comprising:

providing a packet network for carrying a plurality of transmissions from at least one source in which the transmissions from the or each source are divided into a plurality of sequential packets each packet having address data defining an intended address, information data defining information to be transmitted and id data defining a source identity;

the network defining a train of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contain packets and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from the different sources in a sequential arrangement as determined by the network;

monitoring a train of packet transport locations to determine which packet transport locations are empty and which contain a packet;

generating a series of data elements each corresponding to a respective one of the packet transport locations and each identifying whether the respective packet

transport location is empty or whether the respective packet transport location contains a packet;

and carrying out statistical analysis on the series of data elements to determine the characteristics of the transmissions.

Preferably the statistical analysis is carried out in real time so that predictions and characterization of the transmissions can be done in real time.

Preferably, as an alternative, the data elements are stored for example in the hard drive of a PC for subsequent analysis and the statistical analysis is carried out subsequent to completion of the transmissions when a series of transmissions have been recorded for analysis.

Preferably the statistical analysis is carried out at a plurality of different time scales in order to provide the calculations as set out hereinafter. Preferably the statistical analyses are carried out simultaneously by providing for each different time scale a respective one of a plurality of registers and entering information from the data elements into each register sequentially.

Preferably, in a simple single mode, the data elements comprise data bits defining "0" when the respective packet transport location is empty and "1" when the respective packet transport location contains a packet and wherein the information for each register is obtained by adding the contents of a next adjacent previous register.

Using the data bits, the analysis includes, for each register, calculating the sample variance of a set of successive observations of the register contents and from the variances estimating a value of the Hurst parameter H, which is the slope of a line which approximates the behavior of a plot of the logarithm of the variances of the registers versus the values of the sequential indices of the registers. This parameter is known per se but the present algorithm provides a technique for calculating this parameter in real time.

When the packets in the train are provided by a plurality of different sources the data elements are arranged to identify a packet transport location as containing a packet only when the packet is identified from the id data as provided by a selected one of the sources such that the characteristics determined relate to only the transmissions from the selected source.

When the packets in the train are provided by a plurality of different sources the data elements are arranged to identify a packet transport location as empty when no packet from any of the sources is contained and to identify when a packet transport location contains a packet from the id data as provided by each one of the sources which source provided the packet.

The characteristics determined from the data elements can be used to provide a calculation of an effective bandwidth of the transmissions for use in access control and prediction.

In particular the above method can be used in one example for determining whether a source additional to a plurality of existing sources of packet transmissions, each source having a predetermined peak rate of packet transmission, can be connected to a packet network, where a transmission medium of the network has a predetermined maximum allowable peak rate of transmissions. This is preferable effected by the steps of:

carrying out in real time the statistical analysis on the series of data elements to determine an effective bandwidth of the transmissions from the existing sources;

and calculating whether the additional source can be connected by comparing the effective bandwidth, the predetermined peak rate of packet transmission of the additional source and the predetermined maximum allowable peak rate of transmissions of the transmission medium of the network.

As an alternative, the method can be used for calculating whether a source additional to a plurality of existing sources of packet transmissions, each source having a predetermined peak rate of packet transmission, can be connected to a second packet network, a transmission medium of the second packet network having a predetermined maximum allowable peak rate of transmissions. This is effected by the steps of:

connecting the additional source to the packet network;

generating said series of data elements from transmissions from said additional source to said packet network;

carrying out in real time the statistical analysis on the series of data elements to determine an effective bandwidth of the transmissions from said additional source;

and calculating whether the additional source can be connected to the second packet network by using the effective bandwidth of the transmissions from said additional source.

According to a second aspect of the invention there is provided a method for deriving information related to the characteristics of transmissions in a packet network comprising:

providing a packet network for carrying a plurality of transmissions from at least one source in which the transmissions from the or each source are divided into a plurality of sequential packets each packet having address data defining an intended

address, information data defining information to be transmitted and id data defining a source identity;

the network defining a train of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contain packets and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from the different sources in a sequential arrangement as determined by the network;

monitoring a train of packet transport locations to determine which packet transport locations are empty and which contain a packet;

generating information defining which packet transport locations contain a packet and the empty locations therebetween;

and carrying out simultaneously and in real time a series of statistical analyses on the information at a plurality of different time scales to determine the characteristics of the transmissions.

In this aspect, the information is not necessarily the data bits as set forth above but can include other information relating to the population of the packets. However in this aspect, the analysis is effected in real time. Preferably however as explained in detail hereinafter, the information comprises a series of data elements each corresponding to a respective one of the packet transport locations and each identifying whether the respective packet transport location is empty or whether the respective packet transport location contains a packet and wherein the statistical analyses are carried out by providing for each different time scale a respective one of a plurality of registers and entering information from the data elements into each register sequentially.

More preferably the data elements comprise bits which define "0" when the respective packet transport location is empty and "1" when the respective packet transport location contains a packet and wherein the information for each register is obtained by adding the contents of a next adjacent previous register.

According to a third aspect of the present invention the same technique for generating bits relating to the empty and filled packets can be used in reverse in a method of generating packet transmissions for simulating a source having required transmission characteristics for transmitting on a packet network comprising:

providing a packet network for carrying a plurality of transmissions from at least one source in which the transmissions from the or each source are divided into a plurality of sequential packets each packet having address data defining an intended address, information data defining information to be transmitted and id data defining a source identity;

the network defining trains of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contain packets and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from the different sources in a sequential arrangement as determined by the network;

from a statistical analysis of previous actual transmissions on the network, generating for the simulated source the required characteristic;

generating from the required characteristic a series of data elements each corresponding to a respective one of the packet transport locations and each specifying whether, in a simulated train of packets, a respective packet transport location is empty or whether the respective packet transport location contains a packet;

and creating from the data elements the simulated packet train.

Thus as set forth above, a methodology is disclosed for real-time traffic analysis, characterization, prediction, and classification in packet networks. The methodology is based on the simultaneous aggregation of packet arrival times at different time scales. The traffic is represented at the synchronous carrier level by the arrival or non-arrival of a packet. The invention does not require knowledge about the source, nor needs to decode the information contents of the packets. Only the arrival timing information is required. The binary representation of the traffic, that is, the arrival or non-arrival indicator, is processed by a series of processing units and organized in an array of vectors. The processing units are arranged in a series in such a way that the each processing unit operates with the result from the previous unit. The first processing unit operates directly with the timing information. The results of all the processing units are organized in the array of vectors and from these vectors information about the traffic can be derived. Statistical measures on the vectors such as variance, provide appropriate parameters at different time scales to calculate traffic descriptors. These descriptors encapsulate different properties of the traffic, such as burstiness and self-similarity, which were previously not considered in traffic management on packet networks.

A characterization is obtained of the traffic on packet networks suitable for a real-time implementation. The methodology can be applied in real-time traffic classification by training a neural network from calculated second order statistics of the traffic of several known sources. Performance descriptors for the network can also be obtained by calculating the deviation of the traffic distribution from calculated models. Traffic prediction can also be done by training a neural network from a vector of the results of a given processing against a vector of results of the subsequent processing unit; noticing that the

later vector contains information at a larger time scale than the previous. The data produced by the algorithm can also be used for performance prediction.

BRIEF DESCRIPTION OF THE DRAWINGS

One embodiment of the invention will now be described in conjunction with the accompanying drawings in which:

Figure 1 illustrates the algorithm. The timing information from the synchronous line is represented by a '1' if a packet arrival has occurred, and a '0' otherwise. This information is processed and organized in vectors. The different components of the algorithm have been labeled.

Figure 2 shows an application of the algorithm in ATM networks. This is a particular case of the algorithm. The required values of the parameters are shown in the figure. The input information is SONET OC3/OC12, which is a standard synchronous carrier for ATM.

Figure 3 is a diagram showing the first method used to construct the H-meter algorithm input stream of 1's and 0's from ATM capture cell timestamps.

Figure 4 illustrates the second method of re-producing the ATM traffic stream. This method eliminates the first four columns of the array.

Figure 5 describes the data representation process.

Figure 6 describes the data generation process.

Figure 7 represents the data representation process for two channels.

Figure 8 shows a block diagram of the implementation of the algorithm in a parallel processing platform with four processors.

Figure 9 shows a block diagram of an optimized implementation of the algorithm in a parallel processing platform with three processors.

Figure 10 depicts an application of the algorithm in characterization of moving images. The coefficient of variation of the vectors has been plotted for different quality factors of the transmission. The plot shows the logarithm of the coefficient of variation of the vectors against the logarithm of the 2 to the power of the vector number. Using this technique it can be determined how the statistical properties of the traffic vary as a function of the quality level of the video compression.

Figure 11 illustrates an application of the invention in traffic classification. The vertical axis is the log of the variance, and the horizontal axis is the log of the bin size index. The plots represent typical variance-time curves of transmissions of video, audio, and ftp (file transfer protocol) signals. A variance-time curve is a plot of the logarithm of the variance of the vectors against the logarithm of two to the vector number. The plot for each stream is clearly distinct from the others. Since the plots can be constructed in real-

time as the vectors are being calculated, this technique can be applied in real-time to characterize the traffic from measures of the variance-time curves.

Figure 12 illustrates the process of generating the Hurst parameter H from variance time curves. The Hurst parameter is calculated at different time intervals and a plot is constructed with the different values of H. The resolution of the graph depends on the lenth of the time intervals.

Figure 13 shows an application of the algorithm for calculation of the H parameter of the 'jump to hyperspace' scene of the StarWars movie. At the centre of the the plot (left-hand side image) it is seen that the H parameter increases in value as the images changes. This is a motion JPEG encoded transmission of the video over an ATM network.

Figure 14 illustrates the application of the algorithm in classification of different traffic streams. A scene with the character R2D2 from the motion JPEG-coded StarWars movie over ATM is compared agains an audio transmission of the same movie.

Figure 15 shows that application of the algorithm to the observation of scenes of high activity from the motion JPEG-coded StarWars movie.

Figure 16 shows the detrended data as presented to the neural network classifier.

Figure 1 7 depicts a diagram of the competitive learning-based neural network that solves the data classification problem.

Figure 18 shows the after-training weight vectors of the neural network classifier indicated by a 'o', and input vectors indicated by a '+'.

Figure 19 illustrates one method of calculating maximum queue length.

Figure 20 illustrates the method of calculating maximum queue length employed in the estimation of effective bandwidth.

Figure 21 illustrates the calculation of the long term effective bandwidth projection.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Referring to Figure 1, the data is processed and organized by an ordered series of *bins*, and stored in an ordered array of *vectors* in such a way that information can be derived regarding traffic at different time scales. Each bin consists of a primary accumulator and a secondary accumulator. A series of bits representing the presence "1" or absence "0" of a cell or packet in successive transport locations of a traffic stream is fed into the primary accumulator of the first bin of the series. The bins are labelled with the integers {1,2,...,m} and the accumulators are labelled by pairs of integers of the form (i,j) so that the accumulators in bin i are labelled (i,1) and (i,2) for the primary and secondary accumulators (respectively) of bin i. Each accumulator has a number of memory locations and the number of memory locations in accumulator (i,j) is represented by p_{ij}.

When an accumulator receives an integer, it stores the integer in the next available (i.e. empty) memory location if one is available. If all of the memory locations are full, then the accumulator computes the sum of all the integers in its memory locations, outputs the result, clears (erases) all of its memory locations and writes the new integer into the first memory location. When a primary accumulator outputs an integer, it passes it to its corresponding secondary accumulator (i.e. the other accumulator in its bin) and to the primary accumulator of the next bin in the series. When a secondary accumulator outputs an integer, it passes it to the vector which corresponds to its bin (i.e. the vector labeled with the same index).

The array of vectors s_C is used to store the information processed by the bins. Each vector can store n elements, where $n>2$ is a constant.

The bins are configured in two possible ways: a) as shift registers, or b) as recursive registers. The shift register receives a new element always in the first position and shifts down data already stored. The recursive register stores a new element in the position of the array that holds the oldest element by replacing it. In both cases, the oldest element is then sent to the vector s_C for storage. Bin c is cleared every time it has received q_C or $p_C q_C$ elements (depending on the case chosen) and a datum has been calculated.

Similarly, the vectors s_C are configured in two possible ways: a) as an array of shift registers, or b) as an array of recursive registers. In the first case, a new element coming from bin c is stored in the first position of s_C and the other elements are shifted. In the second case, a new element is stored in the position of the oldest element. In both cases the oldest element is dropped.

The algorithm is organized such that the integers passed to the vector s_C of the array are the values of sums of n_C successive bits of the binary stream fed into the primary accumulator of bin 1 where n_C is given by

$$n_c = p_{c2}\Pi_{i=1}^{c} p_{i1}.$$

These numbers thus represent the number of packets arriving in successive time intervals of duration n_C each. We refer to n_C as the *bin count* for bin c.

The vectors s_C contain timing information at different time scales. This information can be used for characterization, prediction, and classification of traffic. In the case of OCn traffic analysis, the traffic can be encoded into a stream of '1's and '0's representing an arrival or non-arrival of a payload cell belonging to a particular VPI/VCI (virtual path identifier/virtual circuit identifier). Timing information of multiple connections is denoted by multiple bits. The time between two consecutive bits (or group of bits) is the minimum cell

spacing time, i.e. maximum rate of the SONET line. This method can also be used to generate traffic from pre-loaded algorithms. This can be very useful for stress-testing switches.

In the special case of this algorithm where $p_{i1}=2$ and $p_{i2}=1$ for each bin i the bin counts are given by $n_c=2^c$. The algorithm starts with a stream of '1's and '0's as input. A '1' represents an occupied ATM cell in a traffic stream. Similarly, a '0' represents an empty cell (no data). The data stream is presented to an accumulator. The first element is simply placed in the primary accumulator of bin 1. When the second element arrives, it is placed in the accumulator , added to the previous element and the sum is stored in an N-element vector (array). The sum is also passed to the primary accumulator for the next bin (column) (which similarly adds every two inputs, stores the sum, and passes it to the next bin) (column). After an accumulator sum is stored, the accumulator is cleared (reset to 0) to await the next input. Since the number of elements (rows) in each vector (column) of the array is limited, when the limit is reached, an insertion pointer is reset to the beginning of the array (so that the next value stored overwrites the previous value). After a certain number of elements (or at the end of processing) each column calculates a variance measure of its array contents using the formula:

$$\mathrm{Var} = \frac{\mathrm{crow}*\mathrm{sumsq}-\mathrm{sum}^2}{\mathrm{crow}*(\mathrm{crow}-1)*4^{\mathrm{colnum}}}$$

where *crow* is the number of elements in the array, *sum* is the summation of the elements of the array, and *sumsq* is the sum of squares of the elements in the array. *colnum* is the identification number of the column (the first column has *colnum*=1, the second has *colnum*=2, etc.).

From the column variances, the Hurst-parameter is found by the formula H = 1-β/2,

where β represents the slope of the variance-time curve on a log-log scale. Thus, the steeper the variance-time curve is, the smaller the corresponding Hurst-parameter value must be.

A first method of constructing the traffic stream is based on the comparison between the timestamp of the most recent cell to the timestamp of the previous cell. This would indicate how much time has elapsed between cells. This interval could then be divided by the minimum cell interval (for 1%OC-3, 0.00028 seconds) to determine how many 1%OC-3 cell periods had elapsed (these would be the number of 0's to insert into the array before inserting a 1. The major drawback of this method is that it would require floating-point division (as fractions of intervals would have to be taken into account). The

resulting implementation missed cells while the processor was busy doing the floating-point operation.

A second method involves no floating-point operations at all. In the algorithm, a '1' or a '0' is input to the accumulator for the first column once every 0.00028 seconds (one 1%OC-3 cell interval time). Since the second column accumulator receives the sum of two numbers in the first, its rate is half of the previous column (it receives an input every 0.00056 seconds). Continuing this to the fifth column, its accumulator receives a new value once for every 32 cell inputs (2^5), so its 'arrival interval' is actually 32 1%OC-3 cell arrival times (0.00896 seconds). When the first cell arrives, its timestamp Ts_1 is noted, and the end of the first interval Ts_end (Ts_1+CPU_clock-ticks_for_2^5_cells) is calculated. The interval cell count is initialized to 1. When the next cell arrives, its timestamp is compared to the end of interval value. If the timestamp is less than Ts_end, the cell has arrived within the interval, and the cell count is incremented. If the timestamp is greater than Ts_end, the interval is over. The interval counter is inserted into the H-meter algorithm (this time at the fifth column) and the counter is then cleared. The next end of interval is calculated as Ts_end+ CPU_clock-ticks_for_2^5_cells. This method counts the number of cells that arrive within a certain time period. It eliminates the work of the first four columns, and involves no floating-point calculations in constructing the traffic stream

Preparation of the Data: A device that prepares the information for the algorithm is a network interface card capable of generating and capturing, in real-time, ATM traffic at OC-3 rates. The card generates and captures traffic using a traffic profile which records the timing information of an ATM traffic stream, without any retention of the specific payload data within the cells. The card is capable of generating and capturing traffic from 3 virtual circuits (VC).

At the heart of the capture/generation card, is an FPGA. The FPGA performs the ATM cell processing tasks, as well as serving as the controller of operation of the card. A microcontroller and an interface to a personal computer serial port are incorporated on the card.

An ATM physical layer device supplies captured cells to the FPGA. This component also allows for ATM cell transmission, if enabled. RAM on the card stores the look-up table for the headers the card is expected to capture. Logic on the card permits it to be plugged into the PCI slot of a PC, allowing a means of storing to disk the captured ATM traffic patterns.

The representation of the ATM traffic stream in the traffic profile file is based on encryption of the ATM cell header; the cell payload is ignored. During traffic capture, the

payload is dropped and for traffic generation, the payload of the outgoing cells is hard-coded with a predetermined value. The encoding scheme allows representation of the headers from number of active VCs (currently 3) as well as that of an idle/unassigned cell. By combining this encrypted traffic profile with a lookup table containing the specific byte values of the headers for each of the active VCs, one is able to reproduce the timing of a traffic stream with specific properties.

For traffic generation, the card is supplied with a desired traffic profile, encoded as described above, and a set of 3 ATM cell headers. As generation proceeds, the traffic profile is decrypted and the specified headers loaded into the outgoing SONET frame. As mentioned previously, the ATM cell payload is not considered, but is fixed to some predetermined value. Since on-board storage of the traffic profile is finite, , either a traffic stream will be fed to the card via a PCI bus or a stored traffic stream will be cycled repeatedly, with a period of approximately half a second for 3 VCs.

Configuration of the board for traffic capture produces a traffic profile file, which is written to disk on a PC. Before capture can begin, a set of three ATM headers must be downloaded to the board for the lookup table. These headers are used for comparison with each header extracted from the incoming SONET frame. A match between a received header and one from the look-up table results in a label being assigned in that cell period. Headers which do not match those in the table are labeled as that of an idle/unassigned cell. In this manner of header extraction, comparison and filtering, and labeling, the timing information from the incoming ATM stream can be recorded.

The card resides in a PCI slot (33 MHz, 5V) of a personal computer. Also required for operation of the card is a connection to a PC's RS-232 serial port. When configured for traffic capture, the captured traffic profile will be written to hard disk at a rate of approximately 0.5 Gbyte/hour. The board's optical driver requires a multi-mode, 1300 nm wavelength, SC-style fiber optic connection.

Implementation for ATM over OC-3: An implementation of the algorithm for where $p_{l1}=2$ and $p_{l2}=1$ is shown in Figure 2. The bin counts are given by $n_c=2^c$. The shift register configuration has been used for the bins and the vectors. The input is SONET information converted into a binary stream. This information is stored and organized in an array of bins and vectors.

The data is processed as follows. Bin 1 receives two bits, adds them up, and sends the result to the vector s_1 and to bin 2. Since bin 2 is not full yet, then two more bits are read into bin 1 and the result sent to s_1 and to bin 2 . This new result is stored in the

first position of s_1 and of bin 2, while the previous values are shifted. Since bin 2 is now full, the two elements are added and the result send to s_2 and to bin 3.

Traffic Characterization: Several calculations can be performed on the vectors s_C to characterize the traffic, for example first order statistical measures like variance, correlation, and the coefficient of variation, or second order statistics like the Hurst parameter H and other fractal dimensions. A methodology using variance-time curves is utilized as follows. The vector v of variances is calculated where each element v_C contains the variance of the elements in the vector s_C. The variances are scaled by dividing each element by the square of bin count. The information is represented by a plot of the logarithm of v versus the logarithm of the bin count. The traffic can then be characterized by parameterization of the variance-time curves. For example, the slope of the variance-time curve is related to the Hurst parameter.

An example of a real-time plot of the coefficient of variation is shown in Figure 10. The curves represent JPEG video traffic at different quality factors. Figure 11 illustrates an application of the invention in traffic classification. The vertical axis is the log of the variance, and the horizontal axis is the log of the bin size index. The plots represent typical variance-time curves of transmissions of video, audio, and ftp (file transfer protocol) signals. A variance-time curve is a plot of the logarithm of the variance of the vectors against the logarithm of two to the vector number. The plot for each stream is clearly distinct from the others. Since the plots can be constructed in real-time as the vectors are being calculated, this technique can be applied in real-time to characterize the traffic from measures of the variance-time curves.

The process which generates the Hurst parameter from variance time curves is illustrated in Figure 12. The Hurst parameter is calculated at different time intervals and a plot is constructed with the different values of H. The resolution of the graph depends on the lenth of the time intervals.

Figure 13 shows an application of the algorithm for calculation of the H parameter of the 'jump to hyperspace' scene of the StarWars movie. At the centre of the the plot (left-hand side image) it is seen that the H parameter increases in value as the images changes. This is a motion JPEG encoded transmission of the video over an ATM network.

Figure 14 illustrates the application of the algorithm in classification of different traffic streams. A scene with the character R2D2 from the motion JPEG-coded StarWars movie over ATM is compared agains an audio transmission of the same movie.

SimilarlyFigure 15 shows that application of the algorithm to the observation of scenes of high activity from the motion JPEG-coded StarWars movie.

The Hurst parameter (H parameter) provides information about the nature of the traffic additional to that which is currently available in the standards. It is a measure of the rate of change of burstiness with respect to changes in time scale. The contribution of the present algorithm for traffic characterization applications is the ability to calculated the H parameter in real-time.

Traffic Classification: The characteristics of different traffic streams can be characterized by variance-time curves. A set of variance vectors v for different traffic streams can be used to train a classifier, for example a neural network. The variances can be calculated in real-time from the array of vectors s_C and the classification can be perform at regular time intervals, for example every second. This information can be used for monitoring.

An example of traffic classification using variance-time curves is shown in Figure 11. The signals represent FTP (top), video (center), and audio (bottom) over ATM. The signals have been characterized by the Hurst parameter H.

The data in its original form, shown in Figure 11, was presented to several network architectures in attempt to solve the classification problem. However, it was determined that the strong linear trends in the time series made it impossible for the network to learn the subtle variations in the data.

To transform the data into a suitable form for presentation to the network, the three time series of Figure 11 were detrended. As the time series appear to be linear, first order detrending techniques were suitable. Two methods of detrending were considered; (i) a least-mean-square fit of a line to the data and (ii) detrending by finding the correlation of each series. Both methods removed the linear trends from the time series.

The least-mean-squares method of detrending is well known. The slope and the intercept of a line that models the data were determined. Using the slope and intercept, the trend component of each element in the series was determined and then removed from the original series, as shown in the equation below. When the data was prepared in this form, it was possible to design a network that could classify the three data types.

$$S_d = S_l - (slope * t_l + intercept)$$

where : S_d = the detrended signal

S_l = the original signal

slope = the slope of the model of the line

t_l = time corresponding to the sample point, i

intercept = the y-intercept of the line that models the data curve

It can be seen from Figure 11 that audio data is highly correlated, video data is less but somewhat correlated and FTP data shows almost no degree of correlation.

Noticing that the three data types could be distinguished by their degree of correlation, a simpler method of detrending is provided in which firstly the slope and the intercept of each time series is calculated. Each time series is then detrended as follows

$$S_d = |S_i - S_{i+1}| + intercept$$

This method finds the correlation of the signal by taking the absolute difference in successive signal amplitudes. This value is then raised by the intercept of the line that models the data curve. When the data is detrended in this manner, it is possible to design a network that could classify the three data types. Signals detrended by this method are shown in Figure 16

Although data detrended by the method of least-mean-squares allowed a network to classify the data, the second method of detrending is less computationally intensive. The second method is equally successful in removing the linear trend information and it involved only addition and subtraction operations whereas the method of least-mean-squares involved a multiplication operation. Reducing the complexity of preprocessing is targeted to classify the data in real-time.

In order to solve this data classification problem, three neural network architectures can be attempted. First, a multi-layer feed-forward network may be considered. The weights are updated using a back propagation algorithm. However, it is not possible to determine a network of this type that could learn to classify the input vectors. Even using momentum, the network's sum-squared error becomes stuck in a local minimum and is never able to reach the error goal.

Next, a LVQ (Linear Vector Quantization) network can be designed. The network consisted of 15 inputs, 3 tan-sigmoidal neurons in the hidden layer and 3 linear output neurons. This network is able to successfully classify the three data types.

Although it is not obvious by observing the input vectors of Figure 16, the three classes of input vectors are linearly separable. This is determined by experimenting with a third network; a competitive network. The competitive network that solves this problem is shown in Figure 17. The network consists of 15 inputs and 3 perceptrons in the output layer. As perceptrons are able to classify the three data types, it is determined that the data must be linearly separable. This network architecture is favored over the LVQ design as it involves fewer weights and neurons. It can been seen Figure 18 that the competitive network found the three data clusters.

Twenty percent of the input vectors are set aside for testing. Once training was completed, the network is tested on the remaining 600 input vectors. After 500 training epochs, the network is able to classify all 600 input vectors into three classes without error.

Traffic Prediction: The array of vectors s_C can be used to predict traffic. A set of vectors s_{i-1} and s_i can be used as a training set for a neural network predictor. Once the neural network is trained it can take the input from a vector s_{i-1} in real-time and predict elements of s_i. This information can be used by different layers of protocols to handle congestion, call admission control, and in general to monitor a source.

Performance Prediction: The following describes a method for estimating the maximum rate of constant bit rate traffic which can be added to a given traffic stream without overflowing a buffer of fixed size during a time window of fixed duration. Let μ denote the rate at which cells are removed from a buffer of size B and let A_{st} denote the number of cells which arrive to the buffer in the time interval (s,t). It is well known that the queue length (i.e. the number of cells in the buffer) at time t is given by $\max_{s \le t} W_{st}$ where $W_{st} = A_{st} - \mu(t-s)$. For example, in Figure 19, suppose that we know W_{st} for every point (s,t) in the plane. The queue length at time t_3 is the maximum value of W_{st} along the line segment AB of Figure 19.

If the maximum queue length is required in the interval $(0, t_3)$, we can simply take the maximum of W_{st} along all the line segments parallel to AB (for example at t_1, t_2 etc.) and then find the largest of those maximum values. However, this is equivalent to just taking the maximum of W_{st} over the entire triangular region ABC. We can envision taking this maximum in a different way, as illustrated Figure 20 We first find the maximum of W_{st} along each of the line segments parallel to CB (for example , the segments labeled $t = s$, $t = s + \alpha_1$, $t = s + \alpha_2$) and then find the largest of those values.

If we find the maximum in this way, we can express the maximum number of cells in the buffer during the time interval (s,t) by the following:

$$M_{st}(\mu) = \max_{v \in (s,t)} \left\{ \max_{\in (s,v)} W_{uv} \right\} = \max_{\alpha \in (0,t-s)} \left\{ \max_{\in (s,t-\alpha)} W_{x,x+\alpha} \right\} = \max_{\alpha \in (0,t-s)} \left\{ \max_{\in (s,t-\alpha)} A_{x,x+\alpha} - \mu\alpha \right. .$$

We now employ a heuristic approximation:

$$\max_{x\in(s,t-\alpha)} A_{x,x+\alpha} \cong m_{st}^{\alpha}$$

where

$$m_{st}^{\alpha} = \max_{j=1,2,\cdots,\left\lfloor\frac{t-s}{\alpha}\right\rfloor} A_{s+(j-1)\alpha,s+j\alpha}$$

and where $\lfloor x \rfloor$ denotes the largest integer not larger than x (i.e. the floor function or truncation) . In words, m_{st}^{α} is defined as follows: Divide the time interval(s,t) into a number of smaller , non-overlapping intervals of length α. Count the number of cells arriving in each of those small intervals and find the maximum of those numbers. The result is m_{st}^{α}. The exact calculation is similar, except that the intervals are overlapping. For example, if $\alpha = 4$, s=0 and t=16, the approximate calculation uses the intervals (0,4),(5,8),(9,12) and (13,16) whereas the exact calculation uses the intervals (0,4),(1,5),(2,6),(3,7),...,(12,15) and (13,16). In the exact calculation $\max_{x\in(s,t-\alpha)} A_{x,x+\alpha}$ we take the maximum value of $A_{x,x+\alpha}$ for all possible values of x (i.e. $x=s,s+1,s+2,...,t-\alpha$). In the approximation m_{st}^{α}, we take the maximum of only the values of $A_{x,x+\alpha}$ corresponding to disjoint intervals $(x, x+\alpha)$ (i.e. $x = s, s+\alpha, s+2\alpha, \cdots, t-\alpha$).

The calculation of the values m_{st}^{α} can be done efficiently in real time using a method similar to that described above for the calculation of the variance-time curve. We employ a series of m bins labelled 1,2,...,m. Each bin is made up of two processing units which can store one integer each. A stream of bits representing the presence "0" or absence "1" of a cell (packet) in each transport location is fed into the first bin. Each bin begins in a cleared state, with no integer stored in either processing unit. If a bin in a cleared state receives an integer, it stores the integer in the first of it's processing units. If a bin which is not in a cleared state (i.e. it has an integer stored in one of it's processing units) receives an integer, it will place the new integer in its second processing unit, calculate the sum of the two integers in its processing unit and pass the sum to the next bin (i.e. bin 1 passes to bin 2, bin 2 passes to bin 3 etc.) . The result of this process is that, at any time, an integer stored in a processing unit of bin *c* represents the sum of 2^{c} consecutive bits fed into the first bin. This in turn corresponds to the number of cells (packets) arriving in a time window of duration 2^{c}. The bin also passes this sum to a maximum filter (described below) and returns to a cleared state, erasing the contents of both processing units.

Associated with each bin is a maximum filter. We thus have m maximum filters labelled 1,2,...m in the same way as the bins . Each maximum filter consists of an input

port, a memory , a counter , an alarm and an output port. Each of these four elements can store one integer. When an integer arrives at the input port of a maximum filter from its corresponding bin, the filter compares the integer in the input port to the integer in memory. The filter then discards the smaller of the two integers, moves the larger of the two integers into its memory, increments the integer in its counter by 1 and compares the counter to the alarm. If the integers match, then the integer in memory is moved to the output port, thus overwriting the integer in the output port, and the integer in memory is set to 0

Associated with each bin is a vector similar to the vectors s_c described in the method for calculating variance-time curves. Each integer which appears at the output port of maximum filter c is passed to vector s_c which records integers passed to it in sequence, dropping the least recent ones when full. If α is an integral power of 2, say $\alpha = 2^c$, then m_{st}^{α} is the maximum of the most recent $\left\lfloor \frac{t-s}{\alpha} \right\rfloor$ integers passed to maximum filter c. In practice, $\left\lfloor \frac{t-s}{\alpha} \right\rfloor$ may be quite large when α is small and so we can not store all $\left\lfloor \frac{t-s}{\alpha} \right\rfloor$ elements at once. This is the reason for the maximum filters. The value of m_{st}^{α} can be obtained by finding the maximum of the most recent $\left\lfloor \frac{t-s}{\delta} \right\rfloor$ integers in the vector s_c where $\alpha = 2^c$ providing that the alarm in maximum filter c is set to $\delta / \alpha = 2^{-c}\delta$. We assume that δ is an integral power of 2, say $\delta = 2^d$, so that the alarm in filter c must be set to $\delta / \alpha = 2^{d-c}$. This effectively breaks up the calculation of m_{st}^{α} by dividing the interval(s,t) into a number of smaller intervals of length δ, taking the maximum over each of the smaller intervals and calculating m_{st}^{α} as the maximum of those.

In this manner we can easily keep track of the values of $m_{t-\Delta,t}^{\alpha}$ for any value Δ of interest for $\alpha = 1, 2, 4, \cdots$ where t represents the current time. We are interested in the maximum number of cells in the buffer in the time interval $(t-\Delta, t)$ which is given by

$$M_{t-\Delta,t}(\mu) = \max_{\alpha \in (0,\Delta)} \{ m_{t-\Delta,t}^{\alpha} - \mu\alpha \ .$$

For each value of c =1,2,...,m we have a value of $m_{t-\Delta,t}^{\alpha}$ where $\alpha = 2^c$. We subtract $\mu\alpha = 2^c \mu$ from this value for each of c=1,2,...,m and find the maximum of the m resulting values. We estimate $M_{t-\Delta,t}(\mu)$ by fitting a quadratic function to a small number of

points of the form $(\alpha, m^{\alpha}_{t-\Delta,t} - \mu\alpha)$ with values of α centred around the integral value of α where the maximum occurred. The operation referred to as "fitting a quadratic function " refers to finding the parameters of a quadratic curve which passes through or as near as possible to the points in question. Methods for fitting a quadratic function are available in the literature.

Note that it is only the last step which depends on the value of μ, the service rate for the buffer. We can thus repeat the last step for a number of different values of μ without having to repeat the measurements involved in filling the vectors s_c.

The effective bandwidth is defined as the minimum value of μ for which the maximum number of cells $M_{t-\Delta,t}(\mu)$ in the buffer does not exceed the buffer capacity B for the duration of the time window of length Δ. We can calculate this quantity at any time by performing a binary search for this value of μ. A binary search consists of evaluating the function $M_{t-\Delta,t}(\mu)$ for two values of μ, say μ_L and μ_R, and continually updating the values of μ_L and μ_R such that the size of the interval (μ_L, μ_R) is decreased by half at each iteration and always contains the value of μ for which $M_{t-\Delta,t}(\mu)$=B. The method of binary search is well known and can be found in the literature.

Projection of Long Term Effective Bandwidth: Let $\mu_{eff}(t,\Delta,B)$ denote the effective bandwidth described above: i.e. $\mu_{eff}(t,\Delta,B)$ represents the minimum value of μ for which $M'_{t-\Delta,t}(\mu) \le B$ where $M'_{t-\Delta,t}(\mu)$ denotes the approximation described above for the maximum number of cells in the buffer during the time window $(t-\Delta,t)$. Suppose that at time t we have been observing a traffic stream for some length of time ω and wish to estimate the bandwidth $\mu_{eff}(t,\Omega,B)$ required to prevent buffer overflow during some time window of length Ω where $\Omega > \omega$. We will assume that $\omega = 2^J\delta$ for some integer J (otherwise we can simply decrease ω accordingly) . Now define $E_k(t,\mu)$ to be a measure of the average value of $M'_{t-\Delta,t}(\mu)$ over the interval of length ω when $\Delta = 2^k\delta$. Specifically:

$$E_k(t,\mu) = 2^{-(J-k)} \sum_{i=1}^{2^{J-k}} M'_{t-i2^k\delta,t}(\mu).$$

In words, we divide the interval of length $\omega = 2^J\delta$ into $2^{(J-k)}$ intervals of length $\Delta = 2^k\delta$ and estimate the maximum queue length over each of these intervals. The average of these values is $E_k(t,\mu)$. In practice we have seen that the variation of $E_k(t,\mu)$ with k can be approximated by the following relation:

$$\log_2[E_k(t,\mu)] \cong a(t,\mu) + kb(t,\mu).$$

Thus if we fix t and μ and plot the logarithm of $E_k(t,\mu)$ as a function of k, we get something that looks close to a straight line. Our method for projecting the effective bandwidth over the long term Ω consists of estimating the coefficients $a(t,\mu)$ and $b(t,\mu)$ by fitting the points $\{(k, \log_2[E_k(t,\mu)])|k=0,1,\cdots J$ to a straight line and then extrapolating a value for $M_{t-\Omega,t}(\mu)$ by extending the line out further. Fitting a straight line consists of finding the line that passes closest to the points and methods for doing this are available in the literature. We thus estimate $M_{t-\Omega,t}(\mu)$ according to:

$$\log_2[M_{t-\Omega,t}(\mu)] \cong a(t,\mu) + \log_2[\Omega]b(t,\mu).$$

Efficient Calculation of $E_k(t,\mu)$: We can calculate the values of $E_k(t,\mu)$ efficiently in real-time as follows. Let m_j^c denote the j th integer passed from maximum filter c to its corresponding vector s_c. If we divide the time interval $(j\delta,(j+1)\delta)$ into intervals of length 2^c then m_j^c is the maximum of the numbers of cells arriving in each of these intervals. With each value of c we associate a column of processors. Each processor has one register which can store an integer, one input and one output. When the register is empty, the processor waits for an integer to appear at its input and stores it in the register upon arrival. When the register is full, the processor waits for the arrival of a second integer at the input. Upon arrival of the second integer, the processor outputs the maximum of the two successive inputs and clears the register. These processors are similar to the bins referred to above except that they output the maximum of the two inputs instead of the sum. In each column of processors, the output of one processor is fed into the input of the next processor in the column. Thus, if successive values of m_j^c are passed to the first processor in the column c of processors, then, at any time, the k th processor in column c contains the maximum of the most recent 2^k values of m_j^c passed to the column of processors. We can think of arranging these processors in a rectangular array with the processors in each column arranged vertically from top to bottom as in Figure 21. We denote the k th processor in column c by P_{kc}. Also, we refer to the collection of the k th processors from all the columns (i.e. $P_{k1}, P_{k2}, \cdots, P_{km}$) as the k th row of processors. At any time t, the k th row of processors contains the values of $m^{\alpha}_{t-2^k\delta,t}$ for $\alpha = 1,2,4,\cdots$. Periodically, each row of processors is polled and the values of $m^{\alpha}_{t-2^k\delta,t}$ are used to calculate $M'_{t-2^k\delta,t}(\mu)$ for various values of μ. These values are in turn fed into row of

processors which output the sum of two successive inputs so that the i th processor in the k th row of this summing array always contains the sum of the 2^i most recent values of $M'_{t-2^k\delta,t}(\mu)$. These provide the value of the sum in the expression for $E_k(t,\mu)$ above if we take $i = J - k$. In this manner we can , in real-time, efficiently keep track of all the quantities necessary to make a long-term effective bandwidth projection.

Call Admission Control: The estimation of effective bandwidth can be used in two ways. In the first way, we have a number of existing sources sending packets into a network. We can estimate the effective bandwidth of the traffic from all of those sources at some point in the network. If another source requests to begin sending additional traffic through the same point in the network the network will reject the request if the sum of the effective bandwidth calculated and a peak bandwidth proposed in the request exceeds the capacity of that section of the network. Alternatively, the network could decrease the effective bandwidth of the existing traffic by allocating more buffer space.

The second application is slightly different. Suppose that a given traffic stream (for example a video-on-demand service) is fed from one location on the network to a multiplicity of other locations on the network. If another location on the network which is not receiving the transmission requests reception of the transmission, a connection will have to be set up from the location providing the service to the location requesting it. In this call setup, the server can specify the equivalent bandwidth to nodes on the network which must carry the traffic from the server to the new recipient. Even if these nodes do not use a protocol which involves the effective bandwidth, the effective bandwidth could simply be sent to them in place of the peak bandwidth and those nodes will set up the connection as if this was the actual peak bandwidth. Alternatively, the server could send both a buffer requirement and a corresponding equivalent bandwidth requirement and the other nodes could react accordingly by making use of a protocol which employs information about effective bandwidth.

CLAIMS:

1. A method for deriving information related to characteristics of transmissions in a packet network comprising:

providing a packet network for carrying a plurality of transmissions from at least one source in which the transmissions from the or each source are divided into a plurality of sequential packets each packet having address data defining an intended address, information data defining information to be transmitted and id data defining a source identity;

the network defining a train of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contain packets and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from the different sources in a sequential arrangement as determined by the network;

monitoring a train of packet transport locations to determine which packet transport locations are empty and which contain a packet;

generating a series of data elements each corresponding to a respective one of the packet transport locations and each identifying whether the respective packet transport location is empty or whether the respective packet transport location contains a packet;

and carrying out statistical analysis on the series of data elements to determine the characteristics of the transmissions.

2. The method according to claim 1 wherein the statistical analysis is carried out in real time.

3. The method according to claim 1 wherein the data elements are stored for subsequent analysis and the statistical analysis is carried out subsequent to completion of the transmissions.

4. The method according to claim 1 wherein the statistical analysis is carried out at a plurality of different time scales.

5. The method according to claim 4 wherein the statistical analyses are carried out simultaneously.

6. The method according to claim 5 wherein the statistical analyses are carried out by providing for each different time scale a respective one of a plurality of registers and entering information from the data elements into each register sequentially.

7. The method according to claim 6 wherein the data elements comprise data bits defining "0" when the respective packet transport location is empty and

"1" when the respective packet transport location contains a packet and wherein the information for each register is obtained by adding the contents of a next adjacent previous register.

8. The method according to claim 7 wherein the analysis includes for each register calculating the sample variance of a set of successive observations of the register contents and from the variances estimating a value of the Hurst parameter H, which is the slope of a line which approximates the behavior of a plot of the logarithm of the variances of the registers versus the values of the sequential indices of the registers.

9. The method according to claim 1 wherein the packets in the train are provided by a plurality of different sources and wherein the data elements are arranged to identify a packet transport location as containing a packet only when the packet is identified from the id data as provided by a selected one of the sources such that the characteristics determined relate to only the transmissions from the selected source.

10. The method according to claim 1 wherein the packets in the train are provided by a plurality of different sources and wherein the data elements are arranged:

to identify a packet transport location as empty when no packet from any of the sources is contained;

to identify when a packet transport location contains a packet from the id data as provided by each one of the sources which source provided the packet.

11. The method according to claim 1 wherein the characteristics determined from the data elements are used to provide a calculation of an effective bandwidth of the transmissions.

12. The method according to claim 1 for determining whether a source additional to a plurality of existing sources of packet transmissions, each source having a predetermined peak rate of packet transmission, can be connected to a packet network, a transmission medium of the network having a predetermined maximum allowable peak rate of transmissions, including the steps of:

carrying out in real time the statistical analysis on the series of data elements to determine an effective bandwidth of the transmissions from the existing sources;

and calculating whether the additional source can be connected by comparing the effective bandwidth, the predetermined peak rate of packet transmission of the additional source and the predetermined maximum allowable peak rate of transmissions of the transmission medium of the network.

13. The method according to claim 1 for determining whether a source additional to a plurality of existing sources of packet transmissions, each source having a predetermined peak rate of packet transmission, can be connected to a second packet network, a transmission medium of the second packet network having a predetermined maximum allowable peak rate of transmissions, including the steps of:

connecting the additional source to the packet network;

generating said series of data elements from transmissions from said additional source to said packet network;

carrying out in real time the statistical analysis on the series of data elements to determine an effective bandwidth of the transmissions from said additional source;

and calculating whether the additional source can be connected to the second packet network by using the effective bandwidth of the transmissions from said additional source.

14. A method for deriving information related to the characteristics of transmissions in a packet network comprising:

providing a packet network for carrying a plurality of transmissions from at least one source in which the transmissions from the or each source are divided into a plurality of sequential packets each packet having address data defining an intended address, information data defining information to be transmitted and id data defining a source identity;

the network defining a train of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contain packets and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from the different sources in a sequential arrangement as determined by the network;

monitoring a train of packet transport locations to determine which packet transport locations are empty and which contain a packet;

generating information defining which packet transport locations contain a packet and the empty locations therebetween;

and carrying out simultaneously and in real time a series of statistical analyses on the information at a plurality of different time scales to determine the characteristics of the transmissions.

15. The method according to claim 14 wherein the information comprises a series of data elements each corresponding to a respective one of the packet transport locations and each identifying whether the respective packet transport location is

empty or whether the respective packet transport location contains a packet and wherein the statistical analyses are carried out by providing for each different time scale a respective one of a plurality of registers and entering information from the data elements into each register sequentially.

16. The method according to claim 15 wherein the data elements comprise bits which define "0" when the respective packet transport location is empty and "1" when the respective packet transport location contains a packet and wherein the information for each register is obtained by adding the contents of a next adjacent previous register.

17. The method according to claim 16 wherein the analysis includes for each register calculating the sample variance of a set of successive observations of the register contents and from the variances estimating a value of the Hurst parameter H, which is the slope of a line which approximates the behavior of a plot of the logarithm of the variances of the registers versus the values of the sequential indices of the registers.

18. The method according to claim 14 wherein the packets in the train are provided by a plurality of different sources and wherein the information is arranged to identify a packet transport location as containing a packet only when the packet is identified from the id data as provided by a selected one of the sources such that the characteristics determined relate to only the transmissions from the selected source.

19. The method according to claim 14 wherein the packets in the train are provided by a plurality of different sources and wherein the information is arranged:

to identify a packet transport location as empty when no packet from any of the sources is contained;

to identify when a packet transport location contains a packet from the id data as provided by each one of the sources which source provided the packet.

20. The method according to claim 14 wherein the characteristics determined from the information are used to provide a calculation of an effective bandwidth of the transmissions.

21. The method according to claim 14 for determining whether a source additional to a plurality of existing sources of packet transmissions, each source having a predetermined peak rate of packet transmission, can be connected to a packet network, a transmission medium of the network having a predetermined maximum allowable peak rate of transmissions, including the steps of:

carrying out in real time the statistical analysis on the information to determine an effective bandwidth of the transmissions from the existing sources; and

and calculating whether the additional source can be connected by comparing the effective bandwidth, the predetermined peak rate of packet transmission of the additional source and the predetermined maximum allowable peak rate of transmissions of the transmission medium of the network.

22. The method according to claim 14 for determining whether a source additional to a plurality of existing sources of packet transmissions, each source having a predetermined peak rate of packet transmission, can be connected to a second packet network, a transmission medium of the second packet network having a predetermined maximum allowable peak rate of transmissions, including the steps of:

connecting the additional source to the packet network;

generating said information from transmissions from said additional source to said packet network;

carrying out in real time the statistical analysis on the information to determine an effective bandwidth of the transmissions from said additional source;

and calculating whether the additional source can be connected to the second packet network by using the effective bandwidth of the transmissions from said additional source.

23. A method of generating packet transmissions for simulating a source having required transmission characteristics for transmitting on a packet network comprising:

providing a packet network for carrying a plurality of transmissions from at least one source in which the transmissions from the or each source are divided into a plurality of sequential packets each packet having address data defining an intended address, information data defining information to be transmitted and id data defining a source identity;

the network defining trains of sequential packet transport locations into which packets are loaded for transmission, such that some packet transport locations in a train contain packets and some packet transport locations are empty and such that, when there is more than one different source, the train contains packets from the different sources in a sequential arrangement as determined by the network;

from a statistical analysis of previous actual transmissions on the network, generating for the simulated source the required characteristic;

generating from the required characteristic a series of data elements each corresponding to a respective one of the packet transport locations and each specifying whether, in a simulated train of packets, a respective packet transport location is empty or whether the respective packet transport location contains a packet;

and creating from the data elements the simulated packet train.

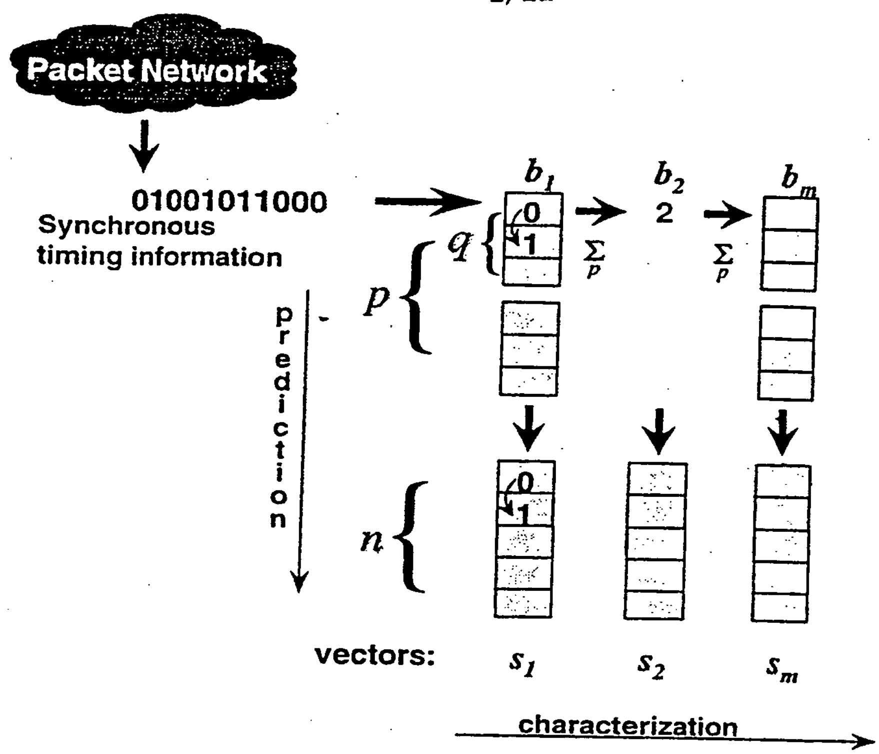


Figure 1

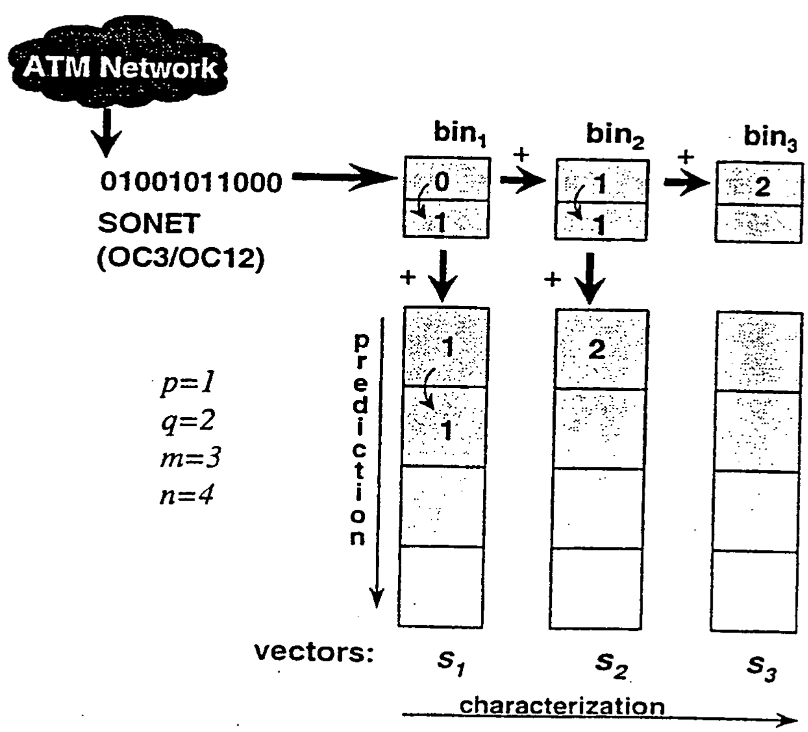


Figure 2

Input to H-meter from ATM Capture Cell Timestamps

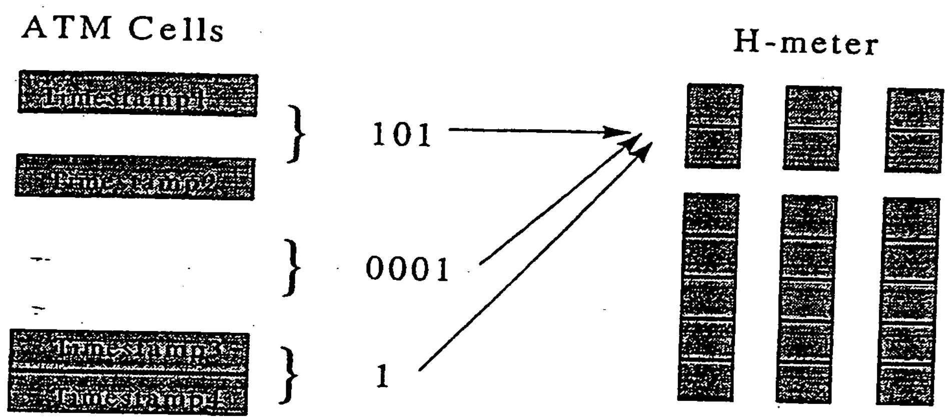


Figure 3

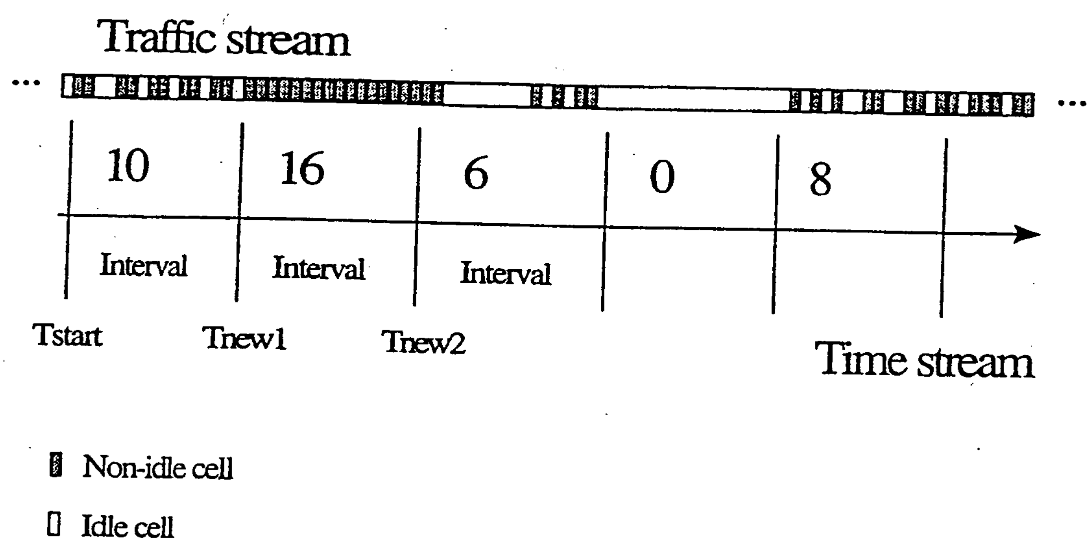


Figure 4

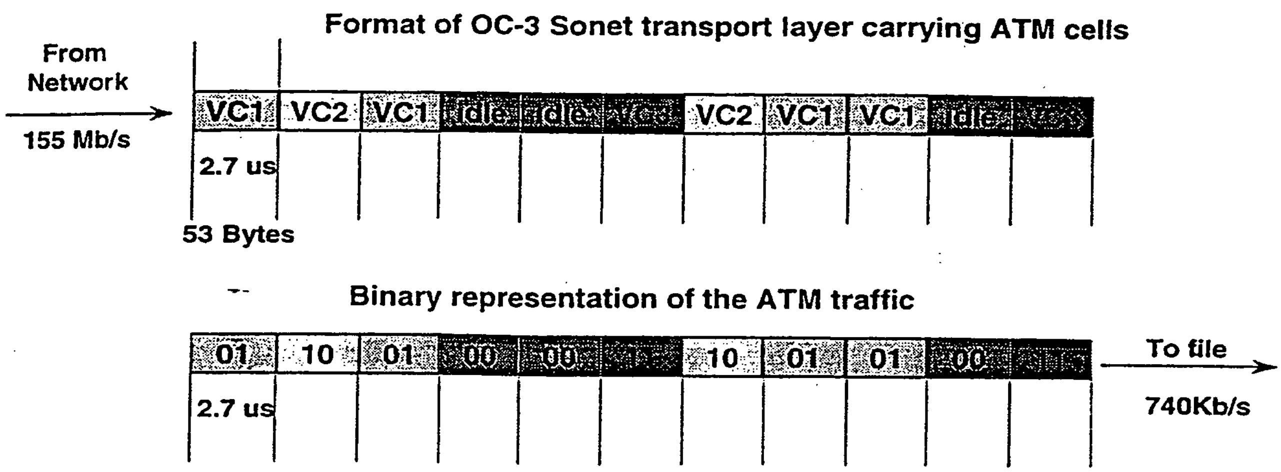


Figure 5

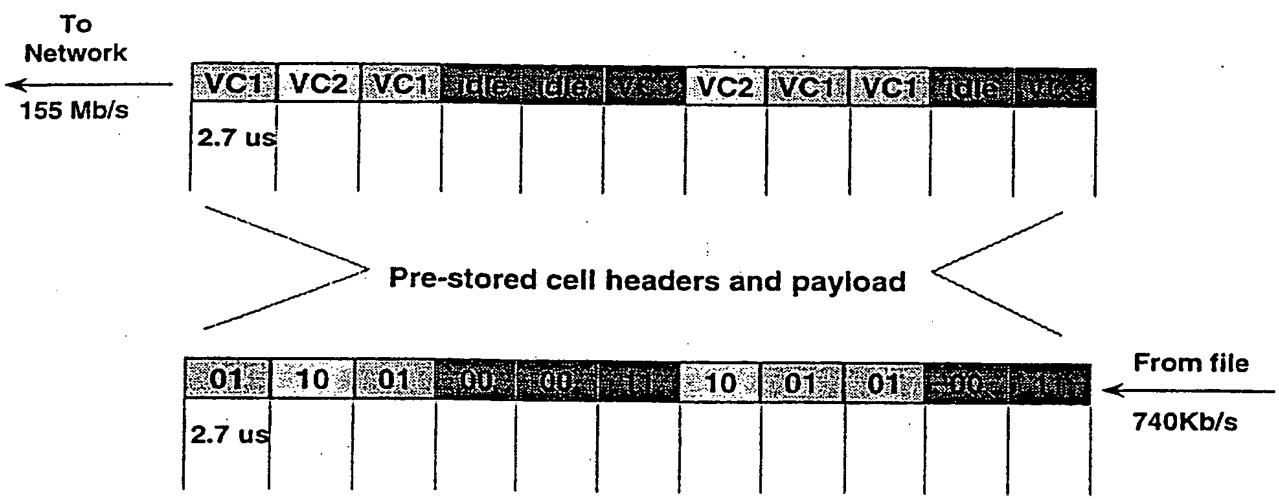


Figure 6

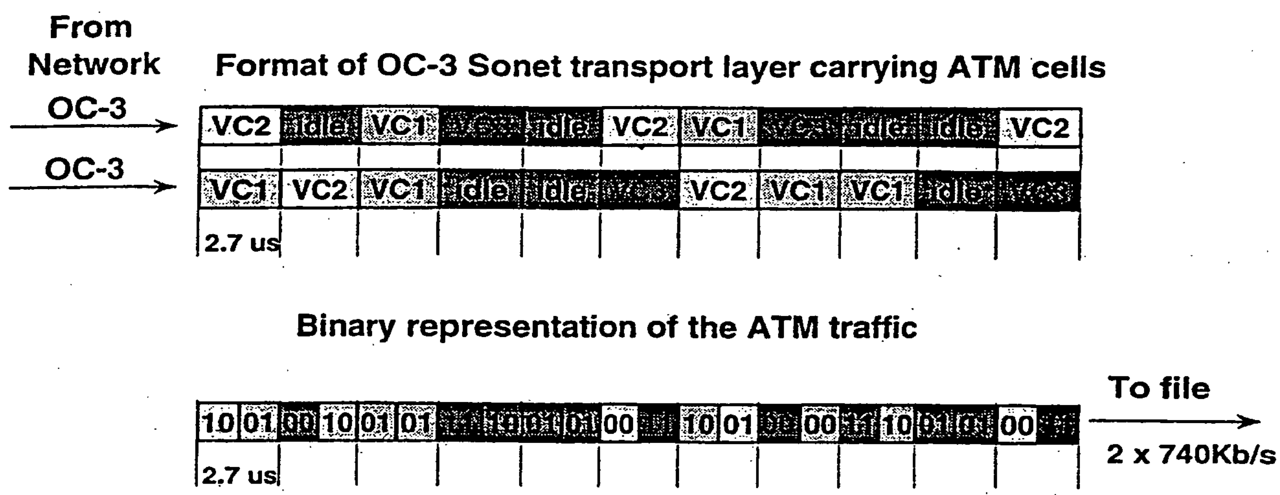


Figure 7

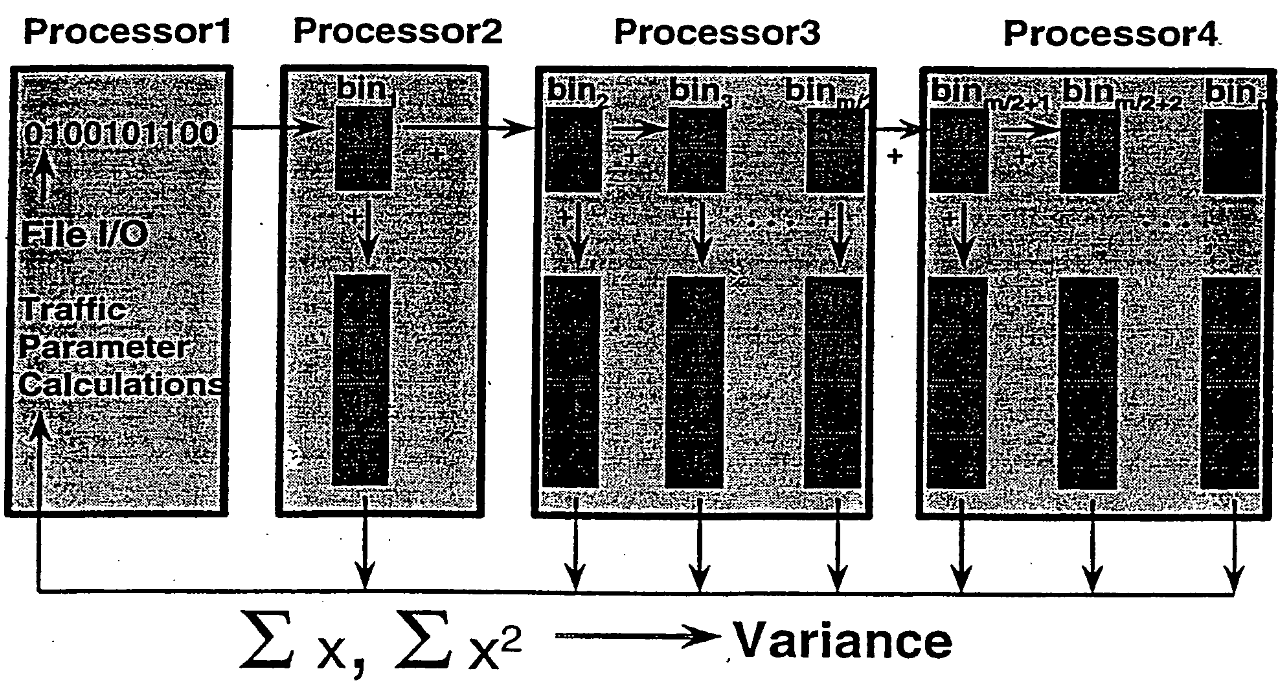


Figure 8

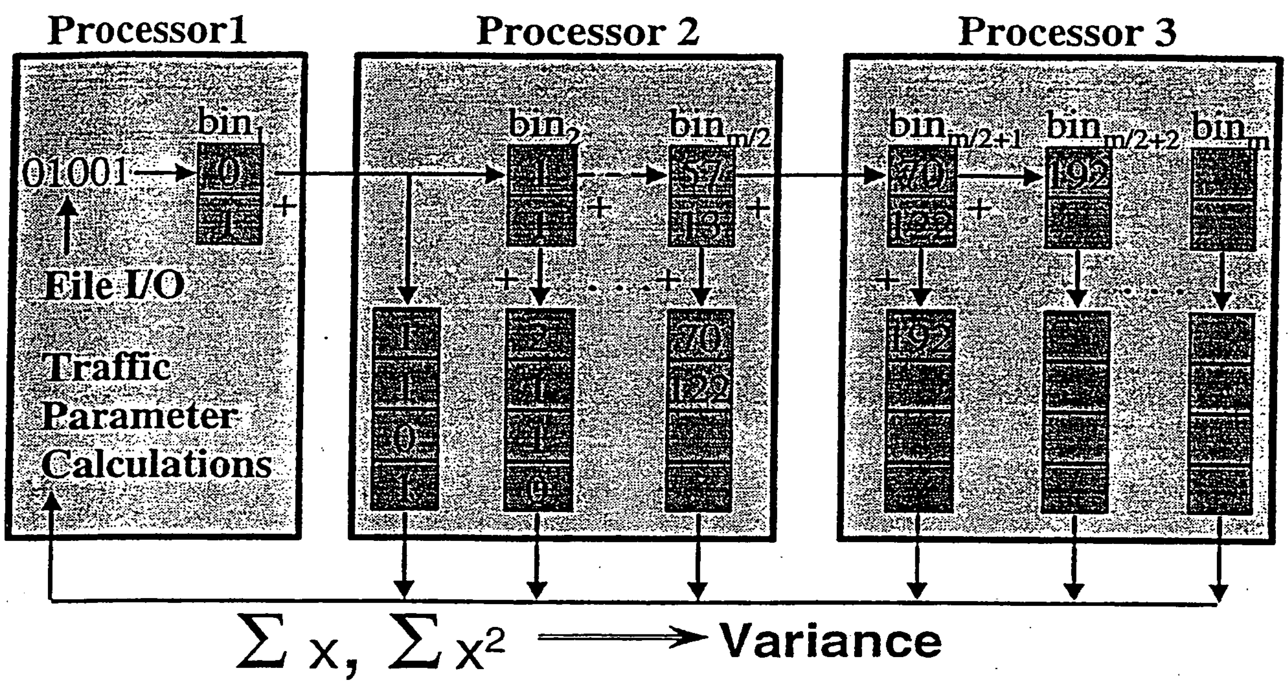


Figure 9

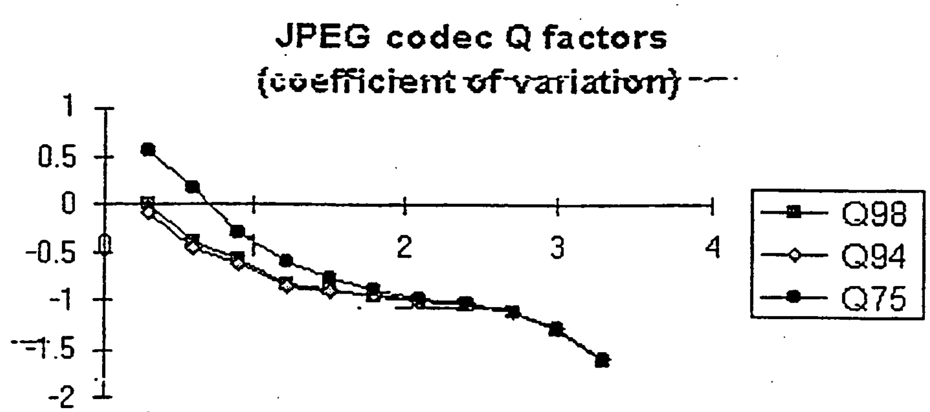


Figure 10

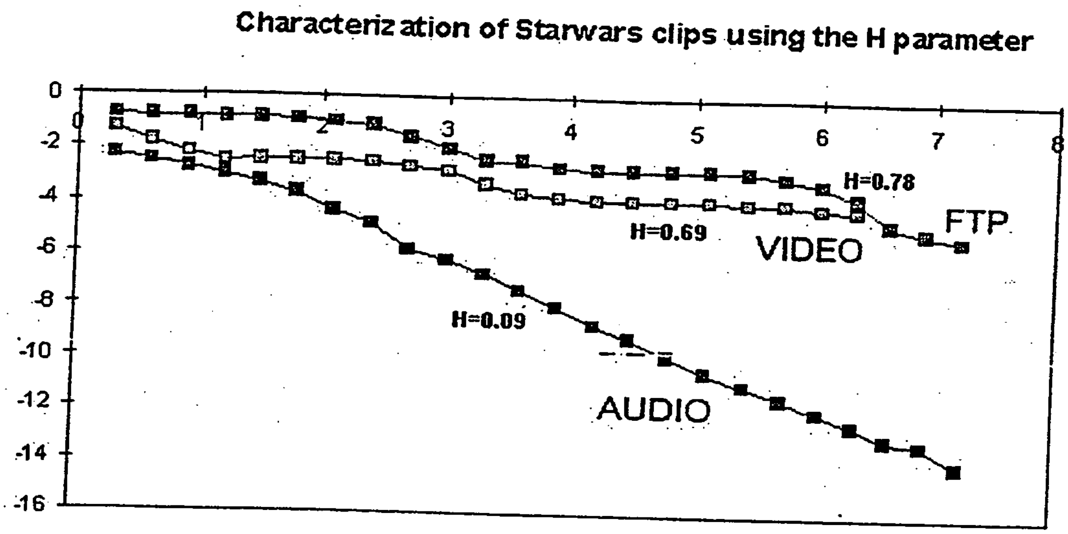


Figure 11

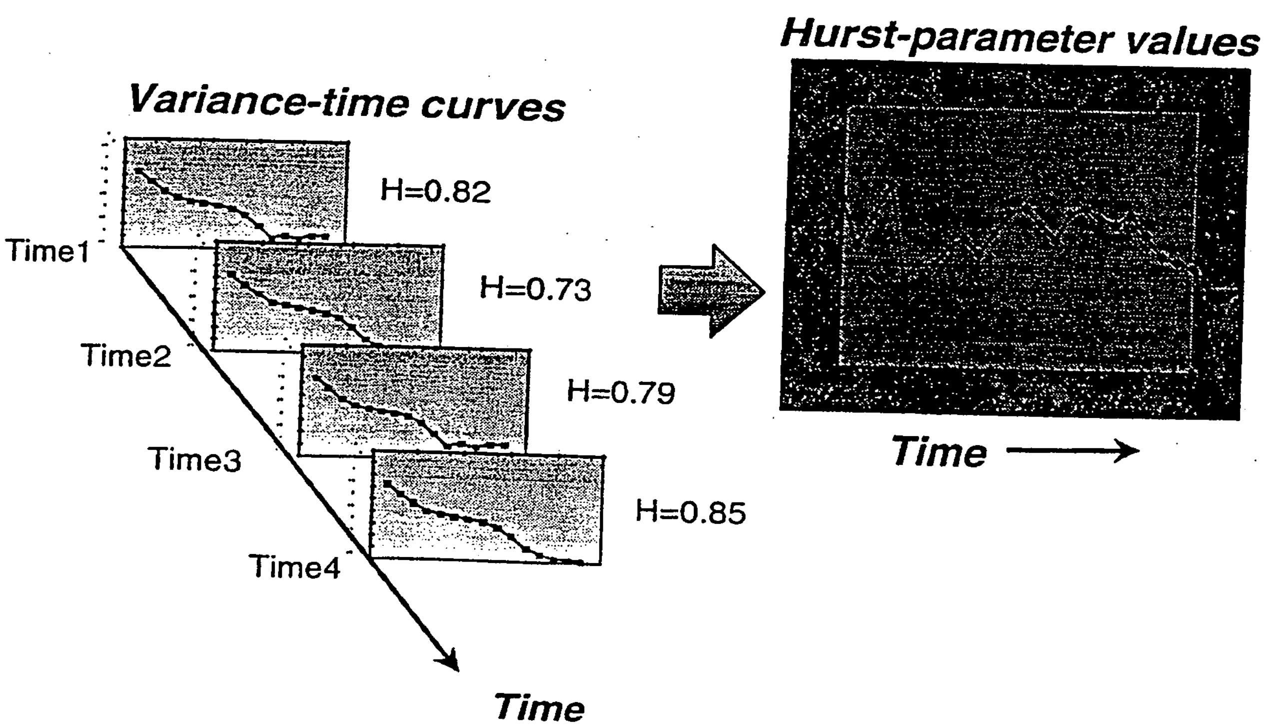


Figure 12

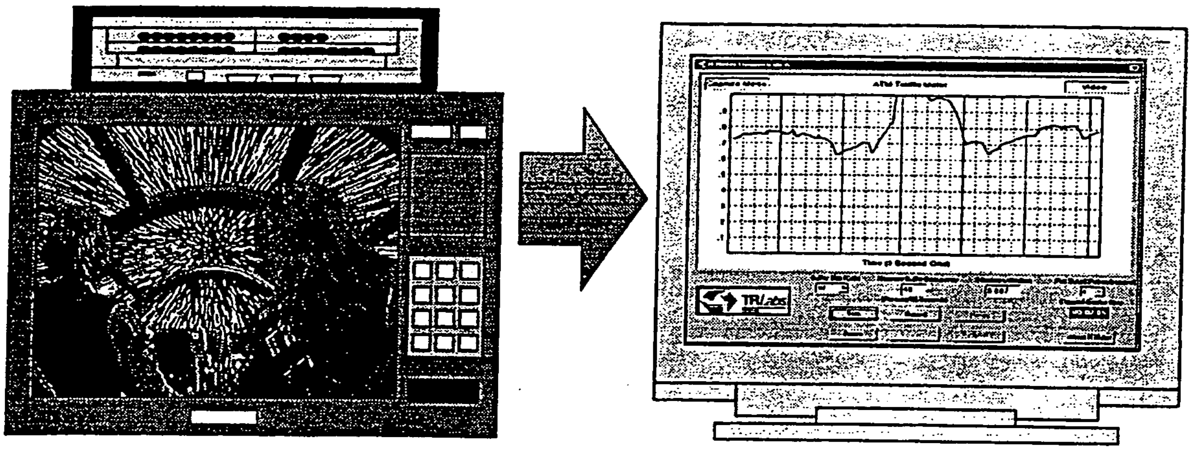


Figure 13

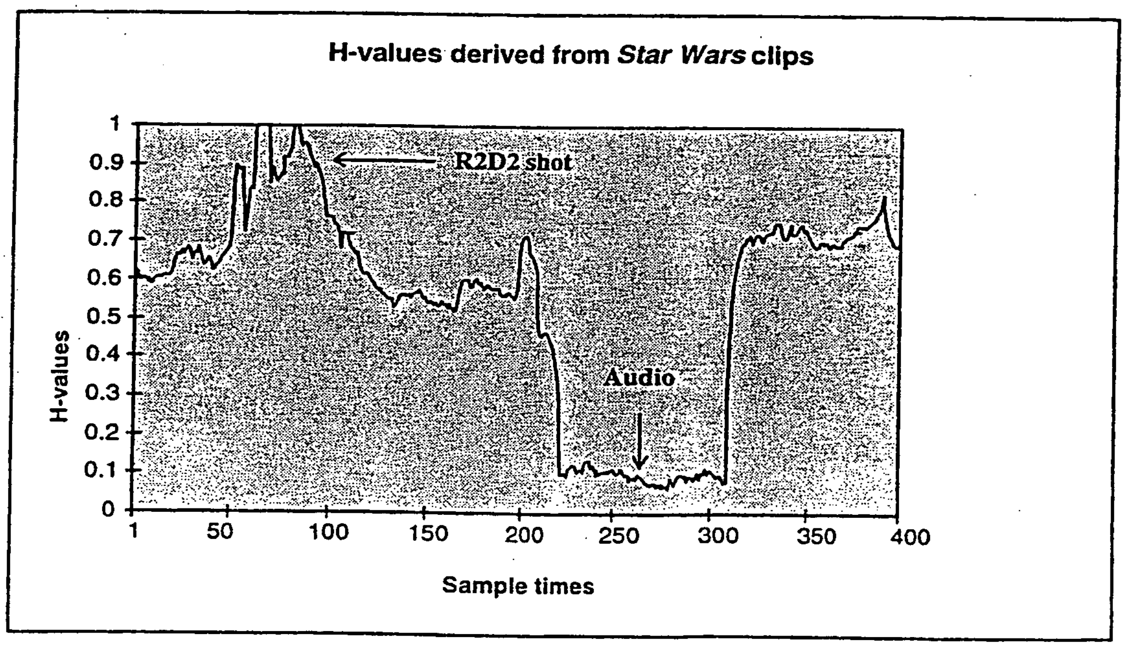


Figure 14

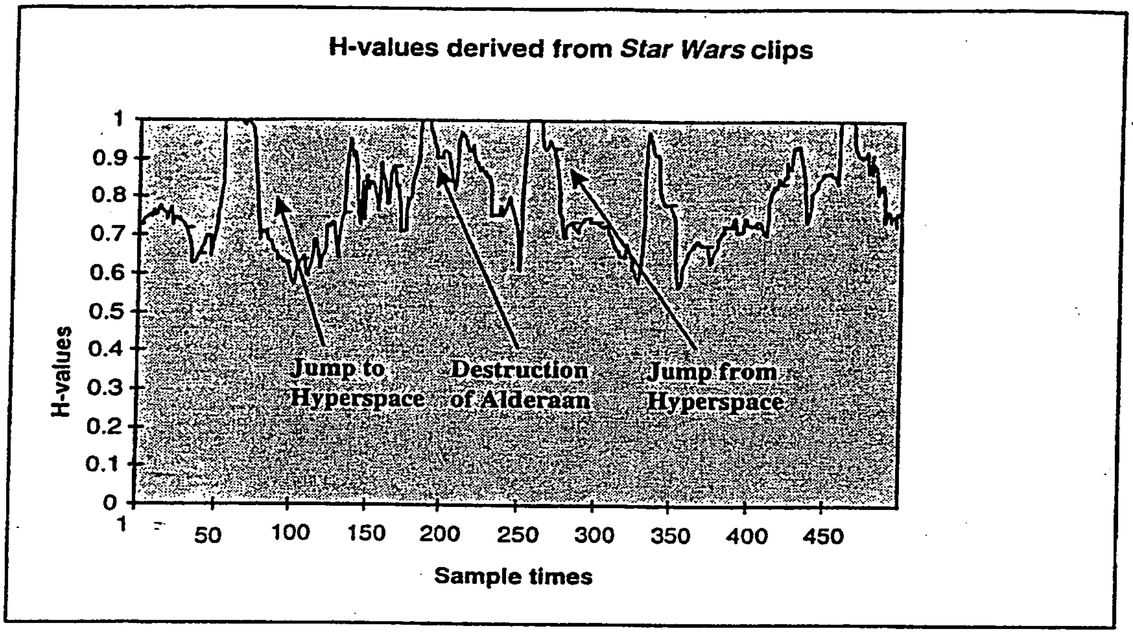


Figure 15

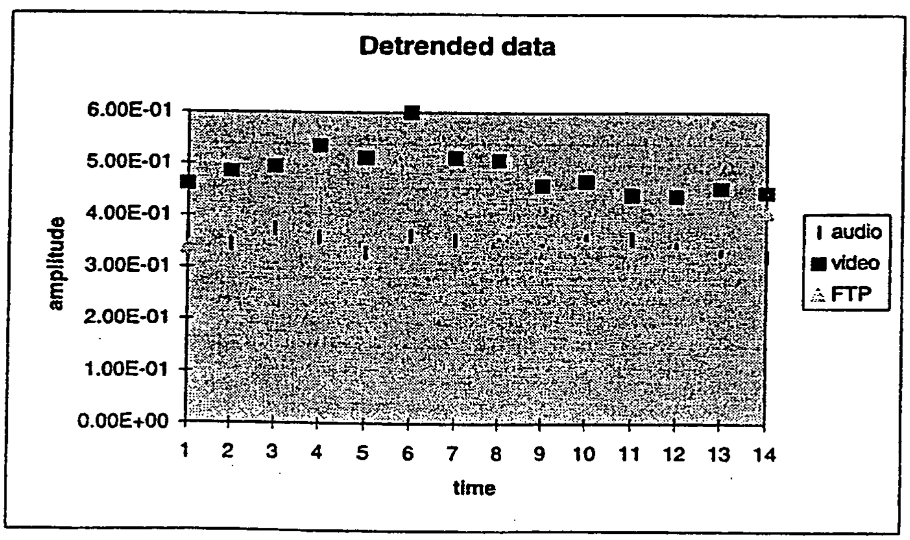


Figure 16

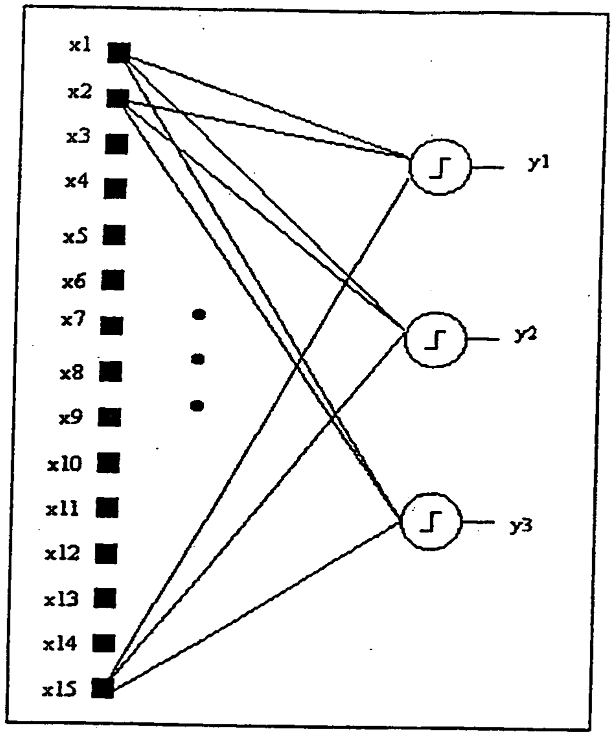


Figure 17

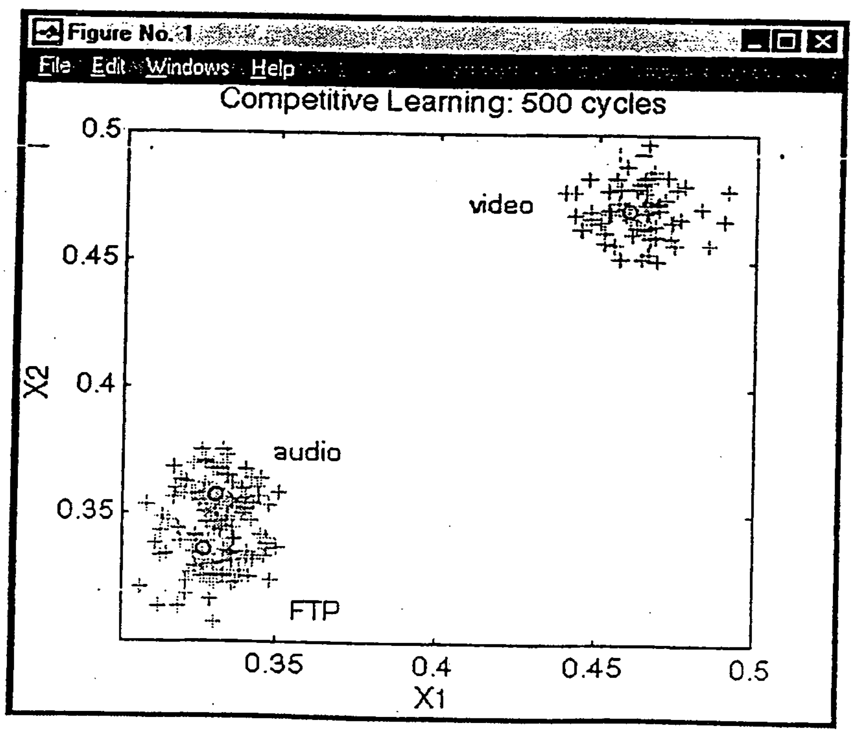


Figure 18

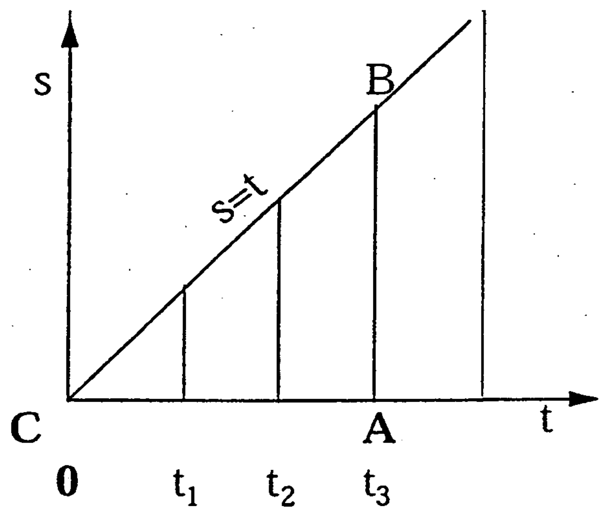


Figure 19

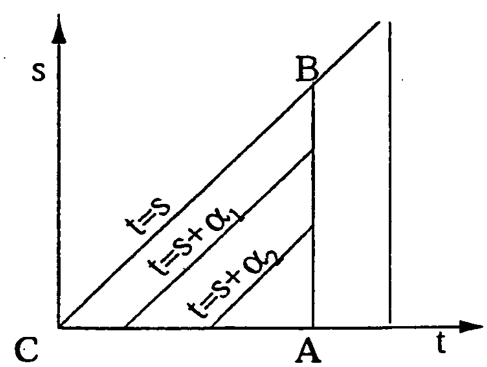


Figure 20

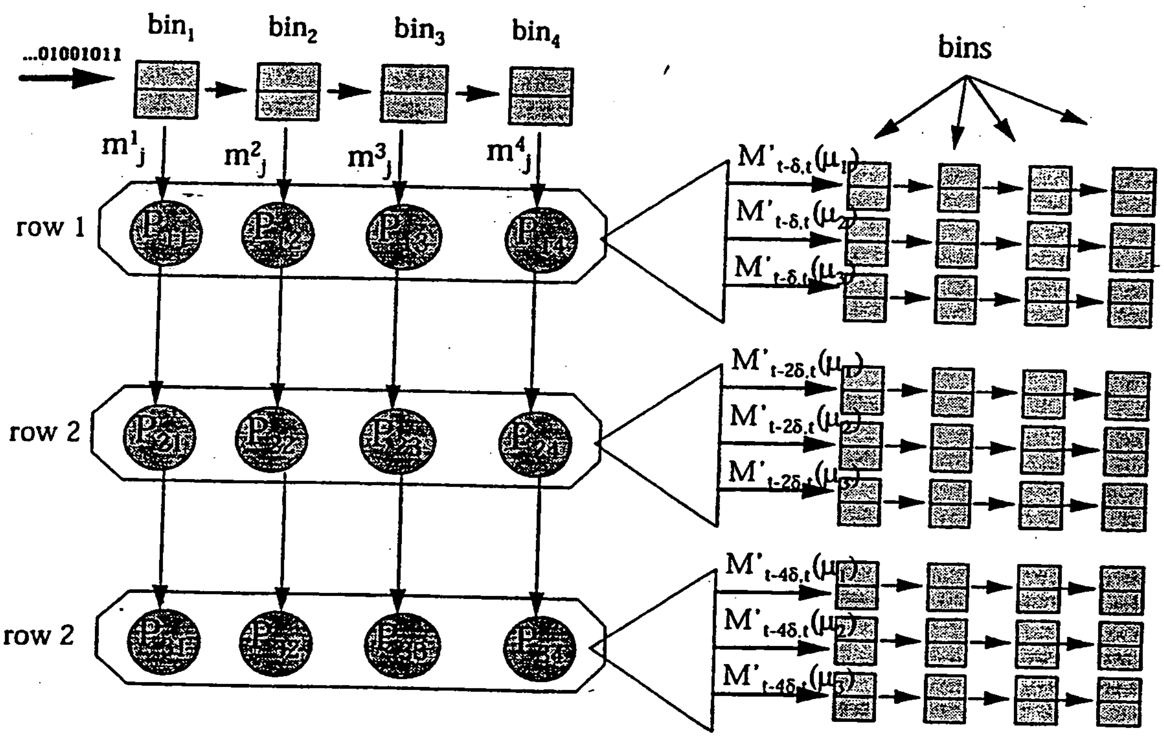

...01001011
bin1
bin2
bin3
bin4
bins
m1j
m2j
m3j
m4j
row 1
row 2
row 2
M't-δ,t(μ1)
M't-δ,t(μ2)
M't-δ,t(μ3)
M't-2δ,t(μ1)
M't-2δ,t(μ2)
M't-2δ,t(μ3)
M't-4δ,t(μ1)
M't-4δ,t(μ2)
M't-4δ,t(μ3)

Figure 21

INTERNATIONAL SEARCH REPORT

International Application No
PCT/CA 97/00991

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04Q11/04 H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04Q H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 357 510 A (NORIZUKI REIKO ET AL) 18 October 1994 see column 3, line 64 - column 4, line 32; claims 1-3 ---	1-3,14, 21,23
A	US 5 357 507 A (HUGHES DAVID A ET AL) 18 October 1994 see column 3, line 29 - column 4, line 19 --- -/--	1,14,23

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 April 1998	07/05/1998

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Veen, G

INTERNATIONAL SEARCH REPORT

Inte: onal Application No
PCT/CA 97/00991

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	DZIONG Z ET AL: "ESTIMATION OF AGGREGATE EFFECTIVE BANDWIDTH FOR TRAFFIC ADMISSION IN ATM NETWORKS" PROCEEDINGS OF INFOCOM '95 - CONFERENCE ON COMPUTER COMMUNICATIONS, FOURTEENTH ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETIES, BOSTON APR. 2 - 6, 1995, vol. VOL. 2, no. CONF. 14, 2 April 1995, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 810-818, XP000580652 see page 811, right-hand column, line 20 - line 47 see page 812, left-hand column, line 18 - right-hand column, line 15 see page 812, left-hand column, line 62 - right-hand column, line 16 see page 812, right-hand column, line 41 - page 813, right-hand column, line 37 ---	1,14,23
A	GIORDANO S ET AL: "A NEW CALL ADMISSION CONTROL SCHEME BASED ON THE SELF SIMILAR NATURE OF MULTIMEDIA TRAFFIC" 1996 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), CONVERGING TECHNOLOGIES FOR TOMORROW'S APPLICATIONS DALLAS, JUNE 23 - 27, 1996, vol. VOL. 3, 23 June 1996, INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS, pages 1612-1618, XP000625082 see page 1614, right-hand column, line 1 - page 1615, left-hand column, line 50 -----	1,14,23

1

INTERNATIONAL SEARCH REPORT

.nformation on patent family members

Inte. onal Application No
PCT/CA 97/00991

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5357510 A	18-10-94	JP 5235980 A	10-09-93
		JP 5268242 A	15-10-93
		JP 5327752 A	10-12-93
		US 5544169 A	06-08-96
		US 5570361 A	29-10-96
US 5357507 A	18-10-94	JP 7283824 A	27-10-95

CORRECTED VERSION*

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q 11/04, H04L 12/56

A1

(11) International Publication Number: WO 98/30059

(43) International Publication Date: 9 July 1998 (09.07.98)

(21) International Application Number: PCT/CA97/00991

(22) International Filing Date: 31 December 1997 (31.12.97)

(30) Priority Data:
60/034,607 3 January 1997 (03.01.97) US

(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application
US 60/034,607 (CON)
Filed on 3 January 1997 (03.01.97)

(71) Applicant (*for all designated States except US*): TELECOMMUNICATIONS RESEARCH LABORATORIES [CA/CA]; 800 Park Plaza, 10611–98 Avenue, Edmonton, Alberta T3K 2P7 (CA).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): RUEDA, Jose, A. [CA/CA]; Unit 10, 31 Carlton Street, Winnipeg, Manitoba R3C 1N71 (CA). BOURLAS, Yair [CA/CA]; 35 Leamington Gate, Winnipeg, Manitoba R3Y 1B8 (CA). DIAMOND, Jeffrey, E. [CA/CA]; 8–812 Wolseley Avenue, Winnipeg, Manitoba R3G 1C6 (CA). McLEOD, Robert, D. [CA/CA]; 38 Elsbury Avenue, Winnipeg, Manitoba R2N 2C4 (CA).

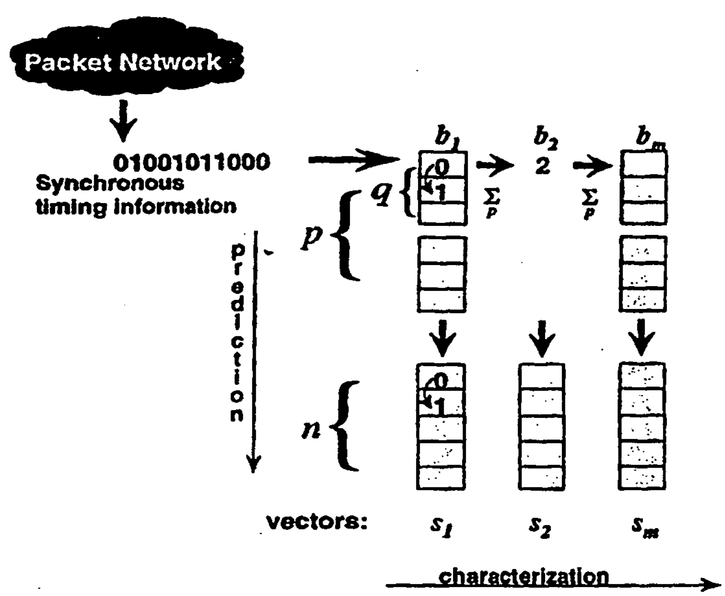
(74) Agent: ADE & COMPANY; Patent and Trademark Agents, 1700 – 360 Main Street, Winnipeg, Manitoba R3C 3Z3 (CA).

(81) Designated States: CA, US, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD FOR REAL–TIME TRAFFIC ANALYSIS ON PACKET NETWORKS

(57) Abstract



An architecture for capture and generation, and a set of methods for characterization, prediction, and classification of traffic in packet networks are disclosed. The architecture consists of a device that stores packet timing information and processes the data so that characterization, prediction, and classification algorithms can perform operations in real–time. A methodology is disclosed for real–time traffic analysis, characterization, prediction, and classification in packet networks. The methodology is based on the simultaneous aggregation of packet arrival times at different times cales. The traffic is represented at the synchronous carrier level by the arrival or non–arrival of a packet. The invention does not require knowledge about the information source, nor needs to decode the information contents of the packets. Only the arrival timing information is required. The invention provides a characterization of the traffic on packet networks suitable for a real–time implementation. The methodology can be applied in real–time traffic classification by training a neural network from calculated second order statistics of the traffic of several known sources. Performance descriptors for the network can also be obtained by calculating the deviation of the traffic distribution from calculated models. Traffic prediction can also be done by training a neural network from a vector of the results of a given processing against a vector of results of the subsequent processing unit; noticing that the latter vector contains information at a larger time scale than the previous. The invention also provides a method of estimating an effective bandwith measure in real time which can be used for connection admission control and dynamic routing in packet networks. The invention provides appropriate traffic descriptors that can be applied in more efficient traffic control on packet networks.

*(Referred to in PCT Gazette No. 38/1998, Section II)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

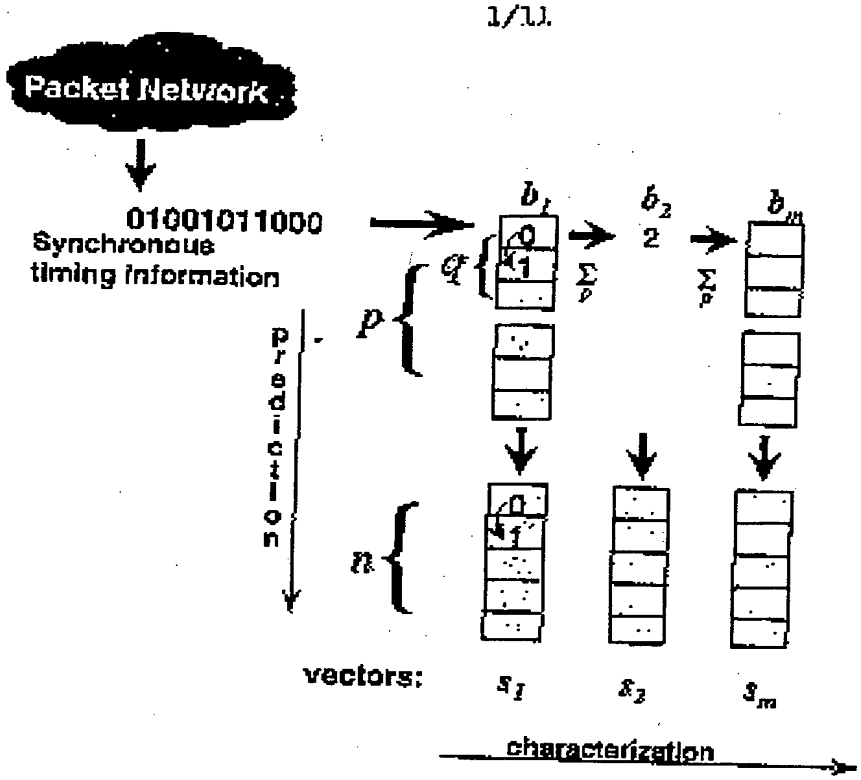


Figure 1

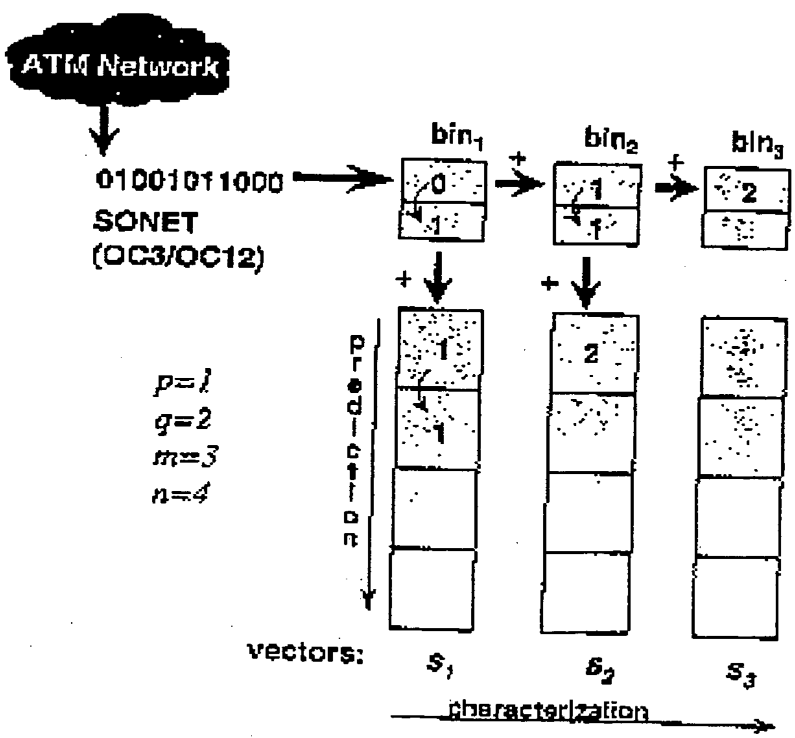


Figure 2

Input to H-meter from ATM Capture Cell Timestamps

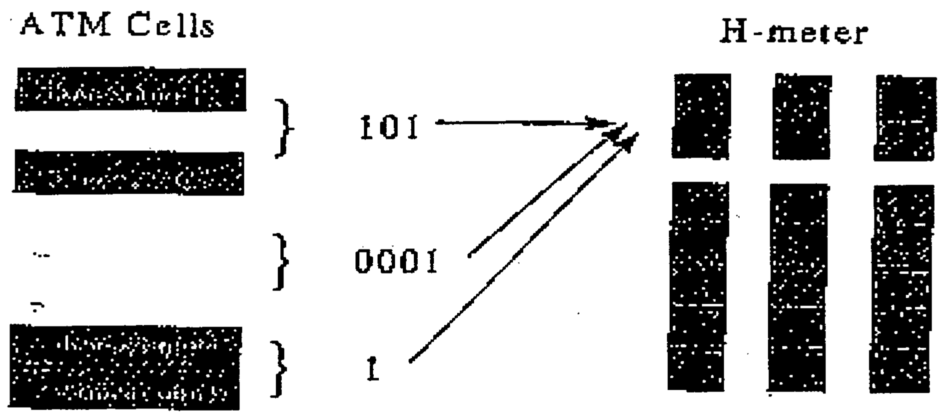
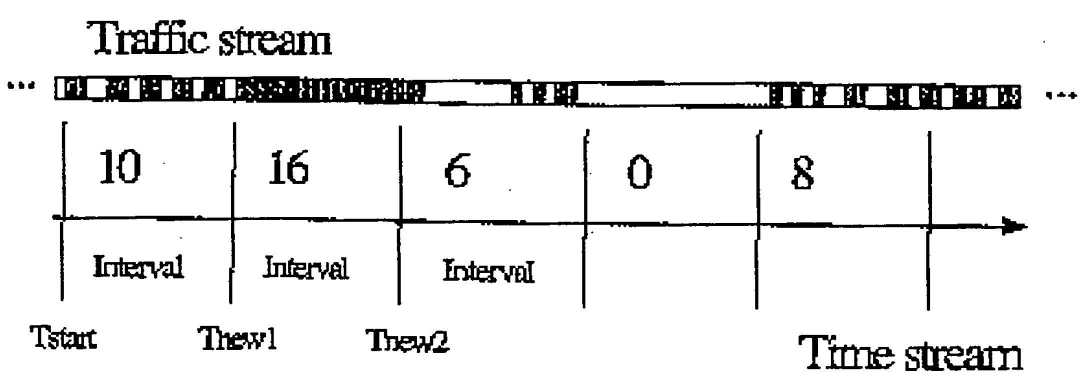


Figure 3



Non-idle cell

Idle cell

Figure 4

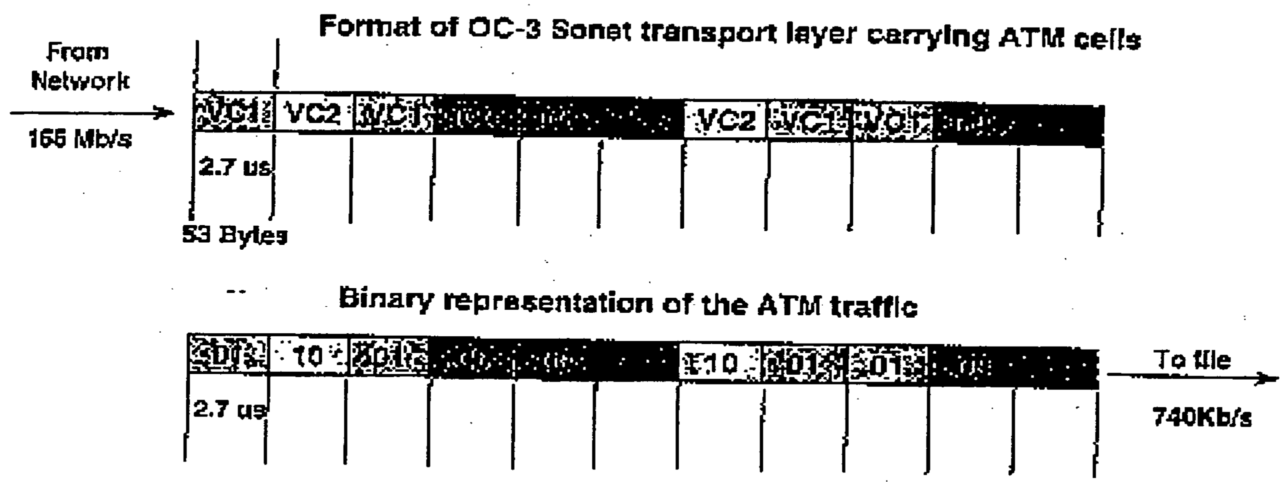


Figure 5

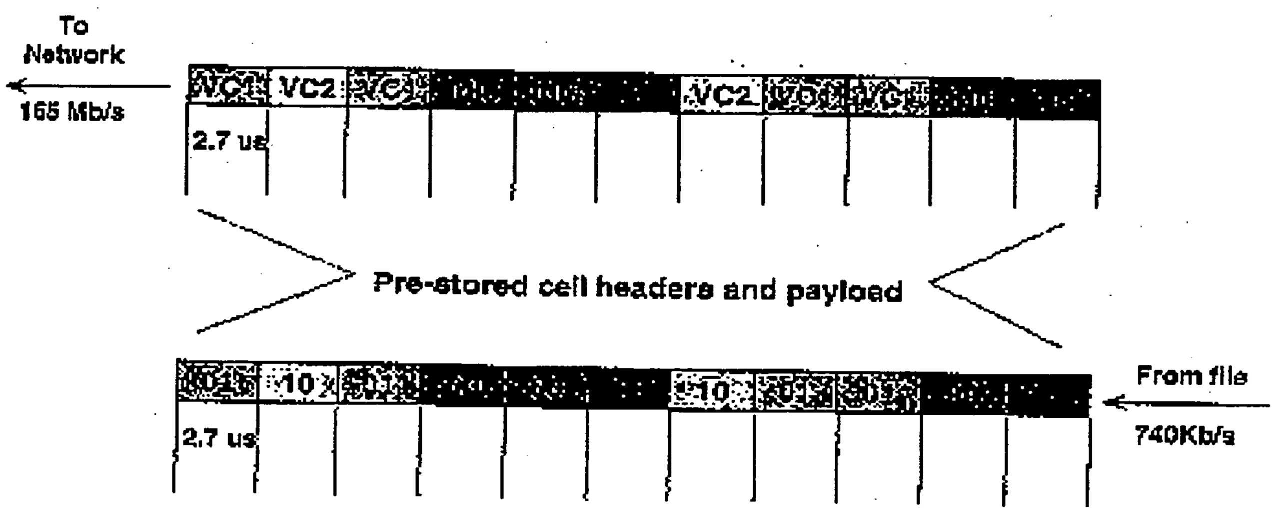


Figure 6

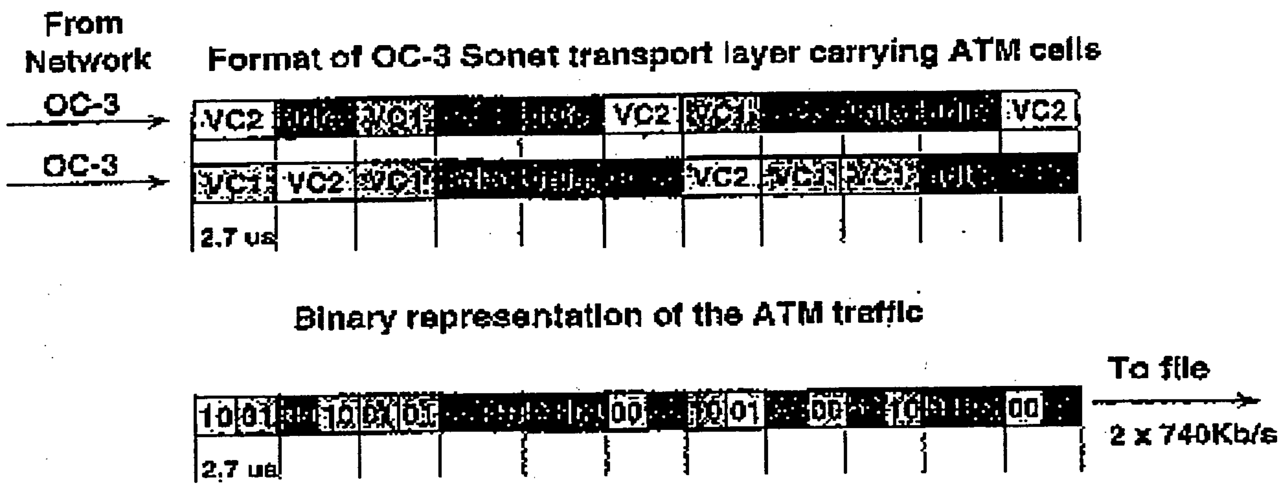


Figure 7

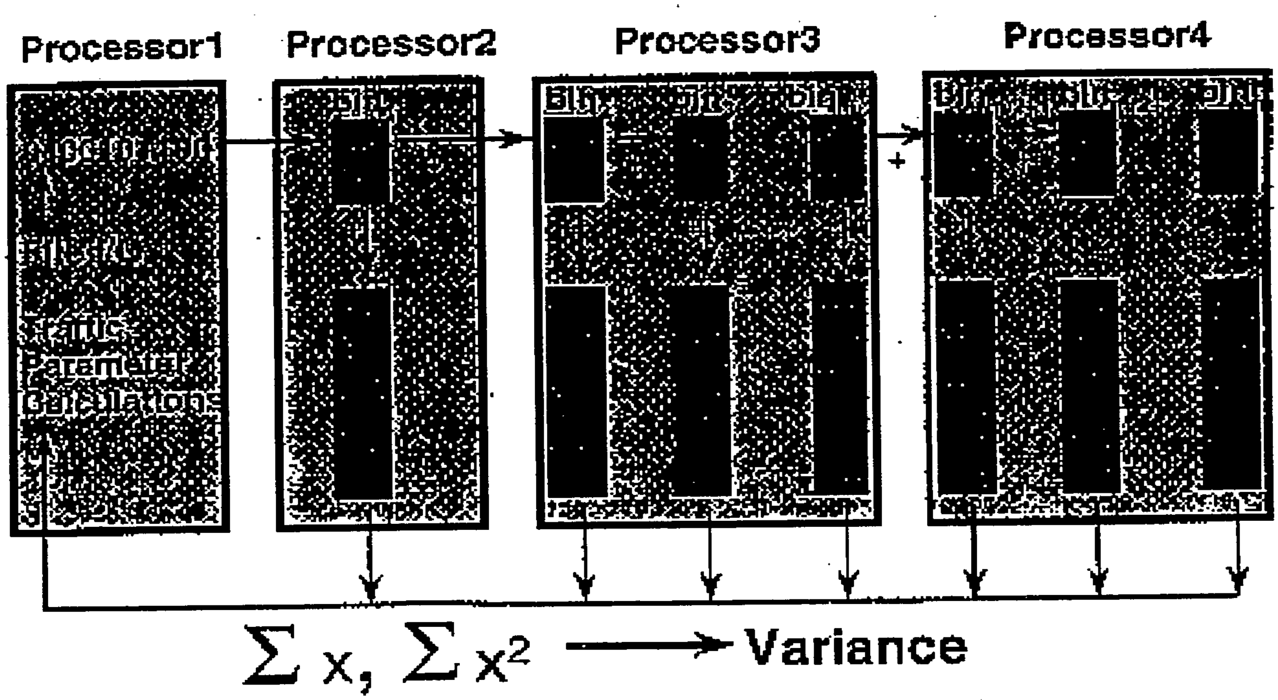


Figure 8

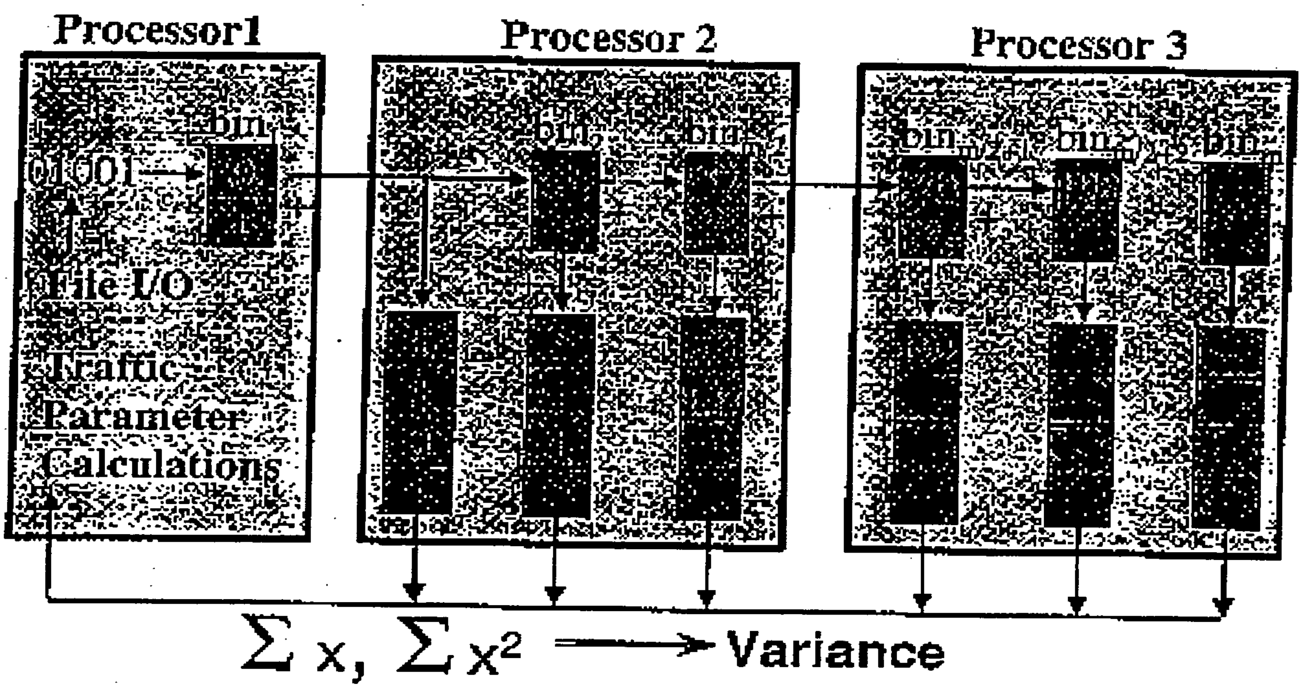


Figure 9

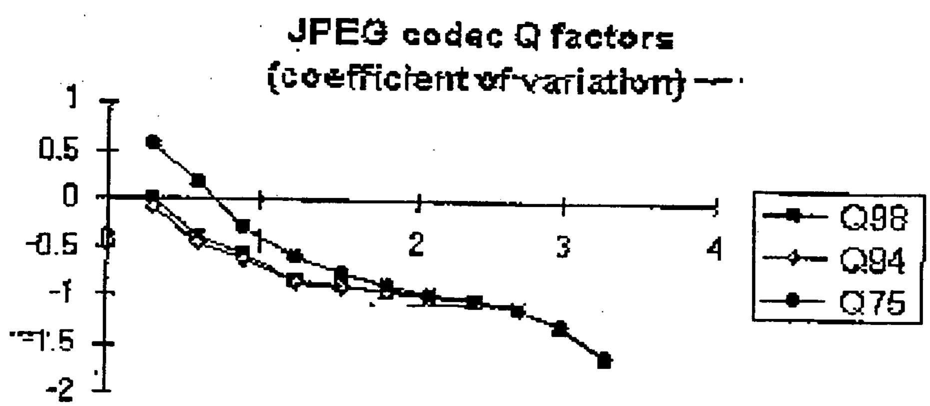


Figure 10

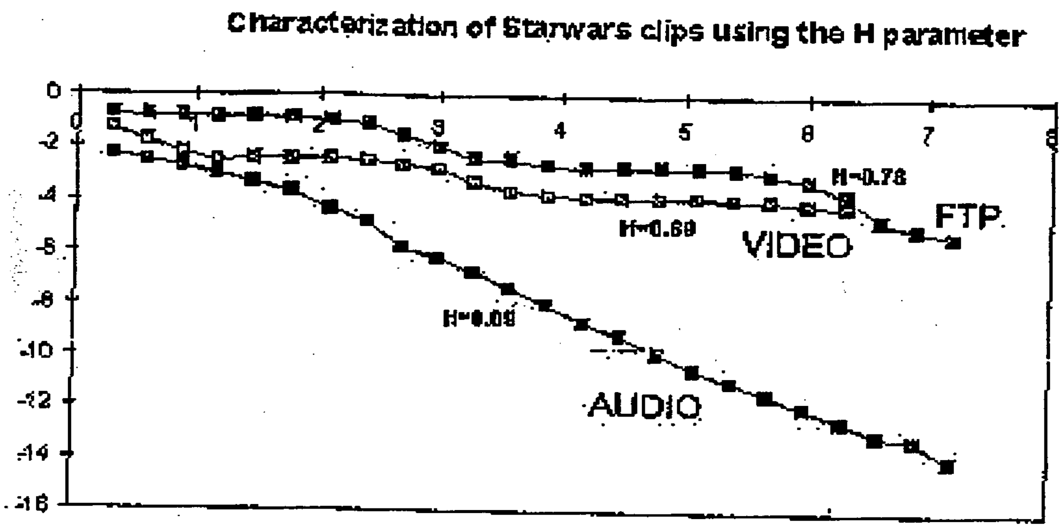


Figure 11

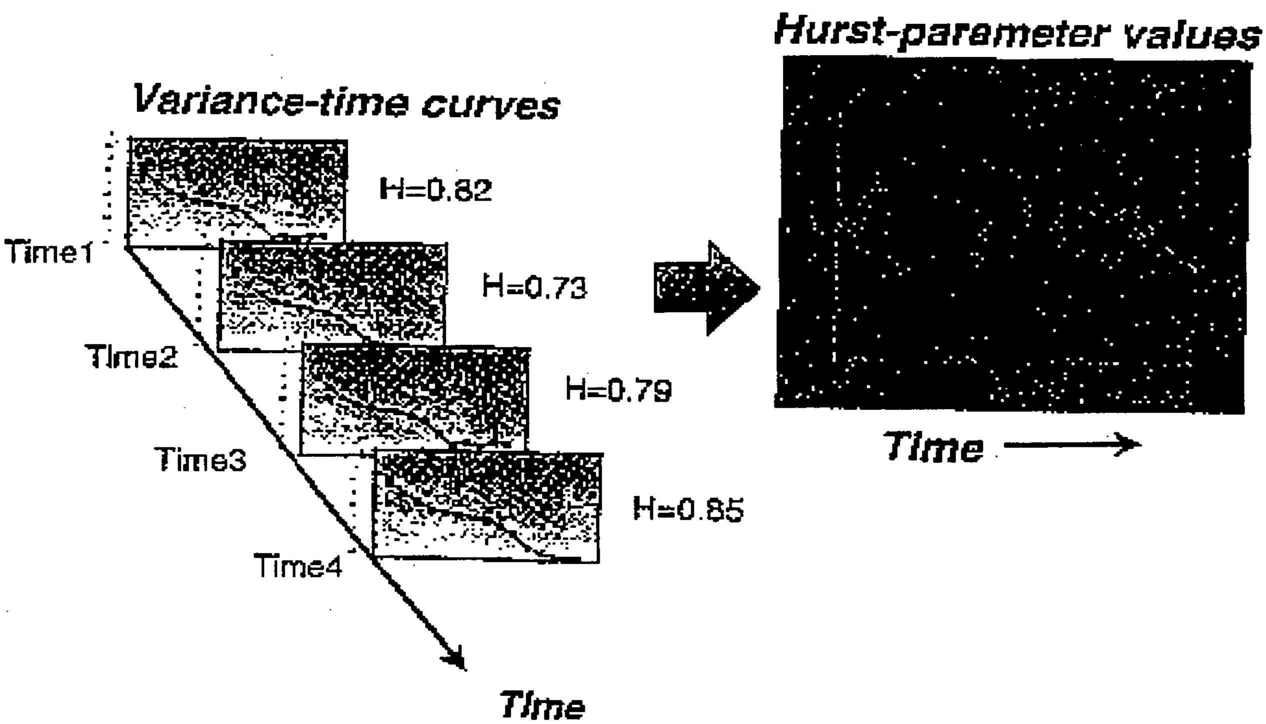


Figure 12

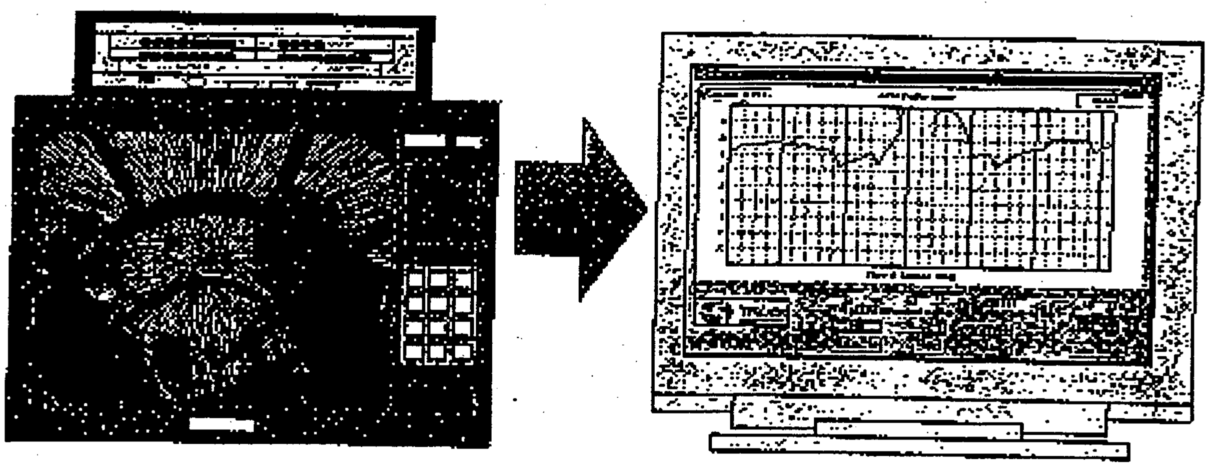


Figure 13

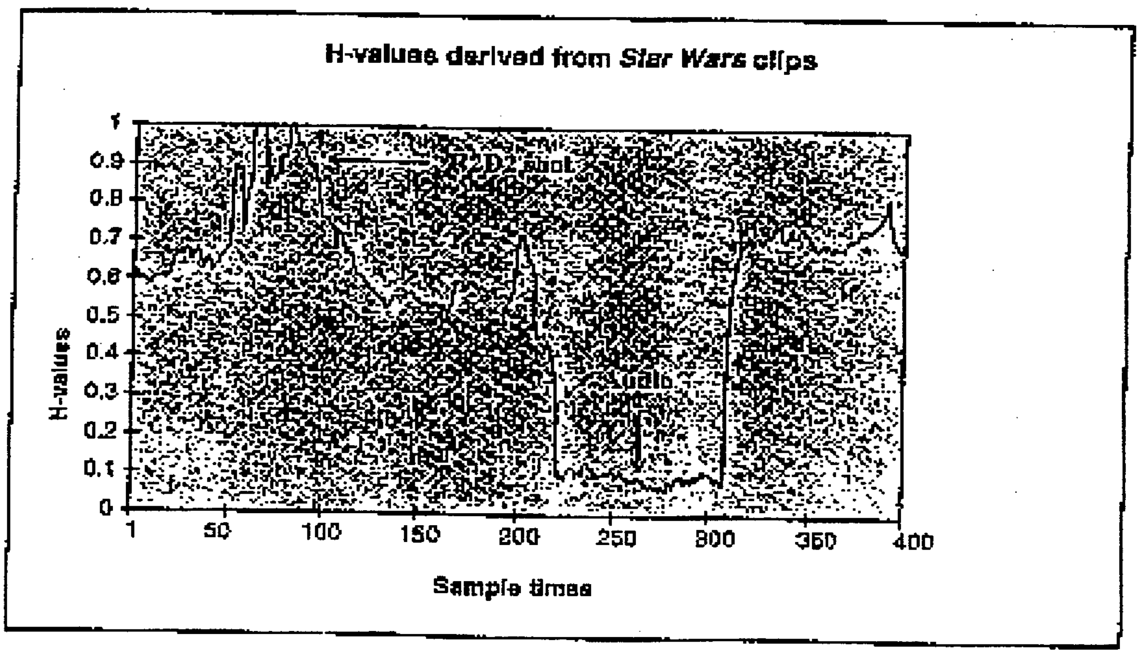


Figure 14

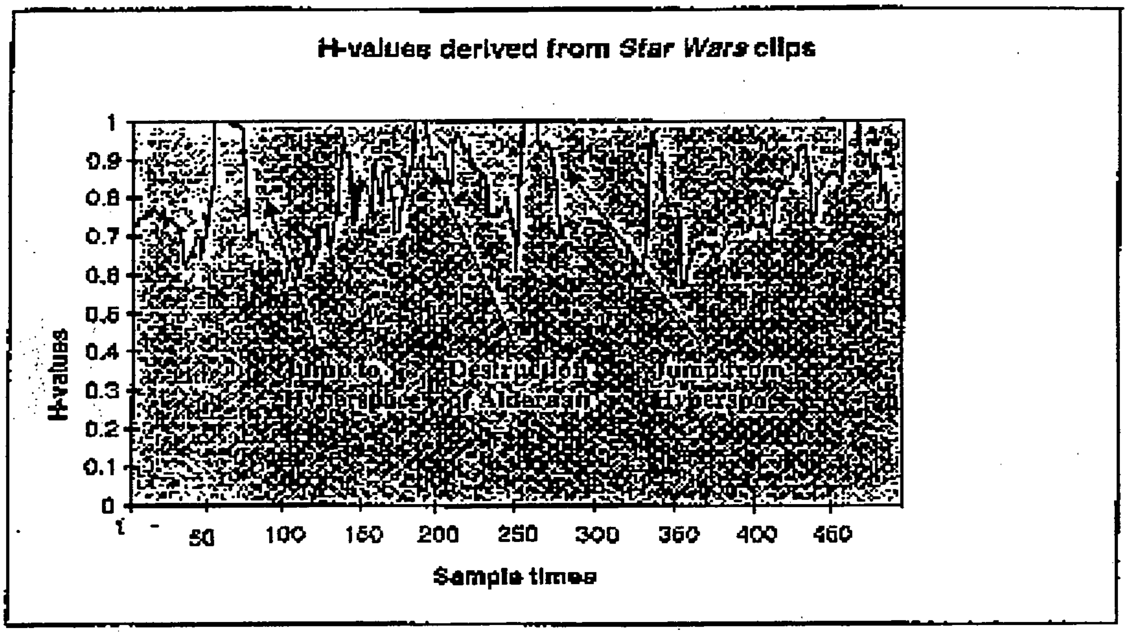


Figure 15

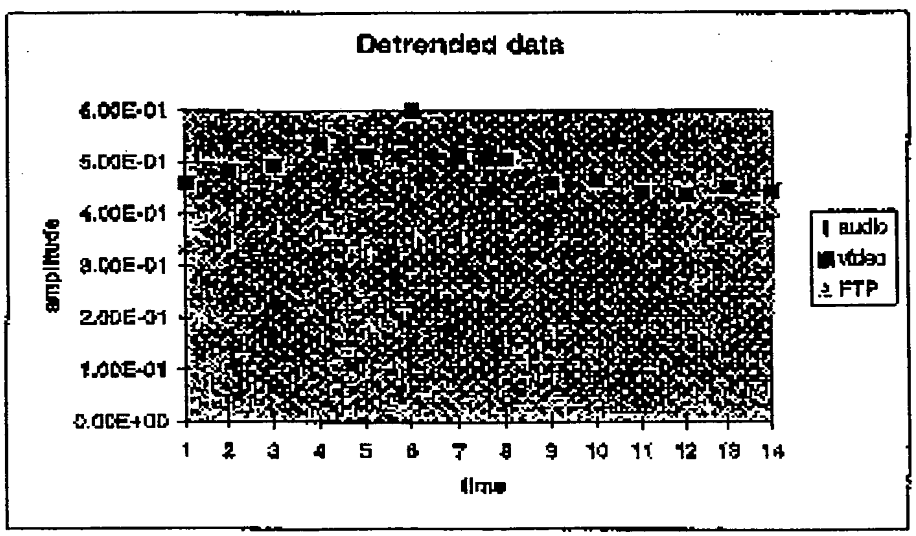


Figure 16

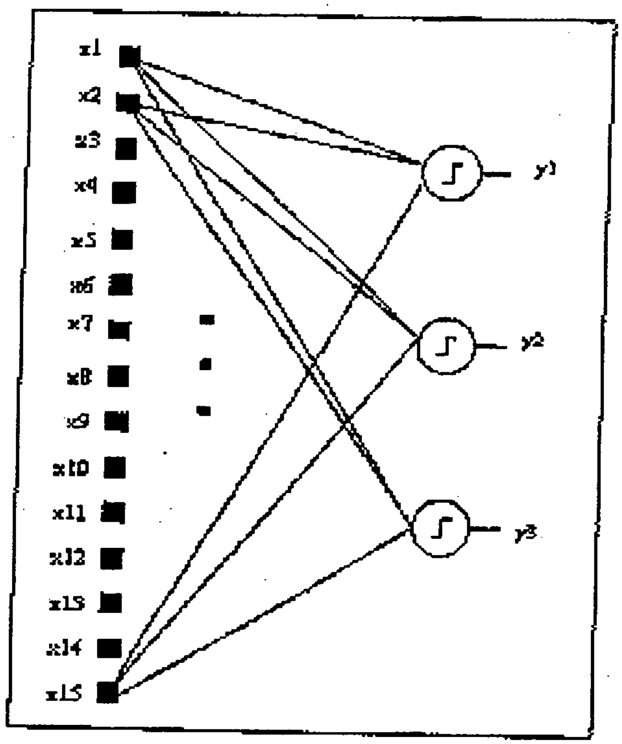


Figure 17

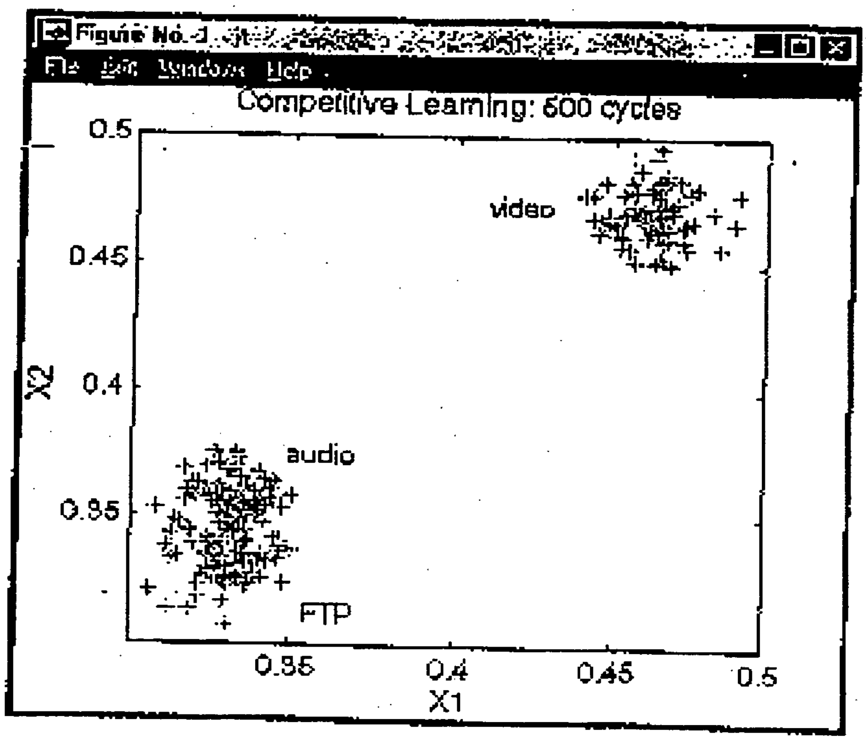


Figure 18

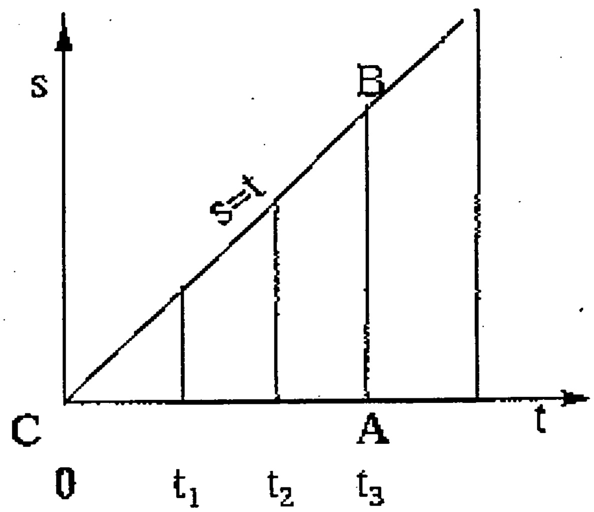


Figure 19

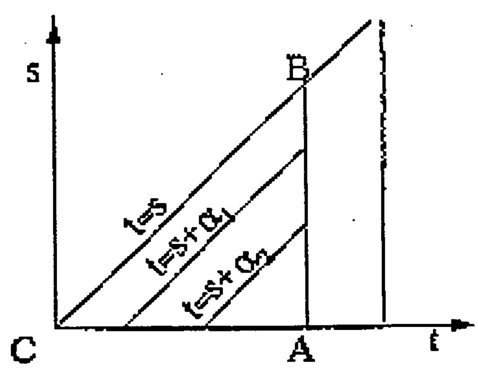


Figure 20

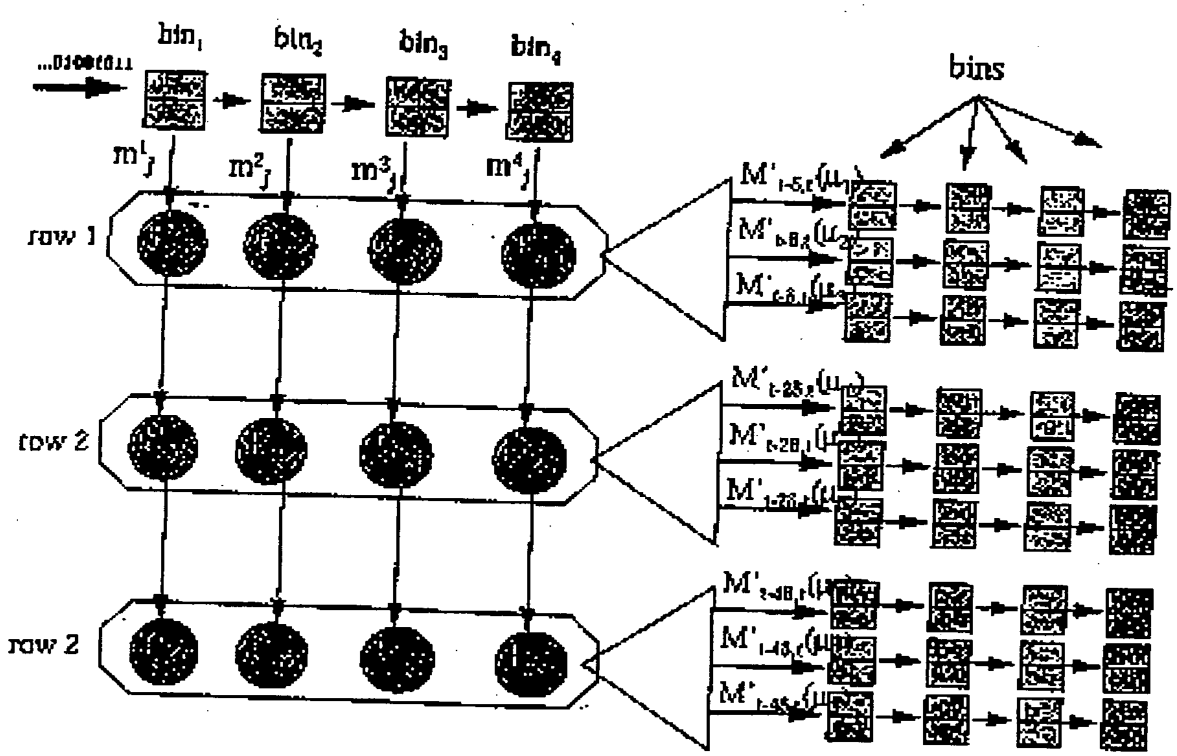


Figure 21